



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**ROBOTICKÁ RUKA S MODELÁŘSKÝMI SERVY**

ROBOTIC HAND WITH RC MODEL SERVOS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**FILIP DOLEŽAL**

**VEDOUcí PRÁCE**

SUPERVISOR

**Prof. Dr. Ing. PAVEL ZEMČÍK**

**BRNO 2018**

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií

Akademický rok 2017/2018

**Zadání bakalářské práce**

Řešitel: **Doležal Filip**

Obor: Informační technologie

Téma: **Robotická ruka s modelářskými RC servy**  
**Robotic Hand with RC Model Servos**

Kategorie: Uživatelská rozhraní

**Pokyny:**

1. Prostudujte způsob řízení modelářských serv a možnosti jejich ovládání počítačem.
2. Navrhněte jednoduchou robotickou ruku sestavenou s pomocí modelářských serv a způsob jejího ovládání.
3. Popište a diskutujte možnosti ovládání navrženého řešení a vyhodnoťte vlastnosti i v závislosti na použitých servech.
4. Navrženou robotickou ruku implementujte a demonstруйте její funkčnost.
5. Diskutujte dosažené výsledky a možnosti pokračování práce.

**Literatura:**

- Dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zemčík Pavel, prof. Dr. Ing., UPGM FIT VUT**

Datum zadání: 1. listopadu 2017

Datum odevzdání: 16. května 2018

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
Fakulta informačních technologií  
Ústav počítačové grafiky a multimédií  
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Jan Černocký  
vedoucí ústavu

## Abstrakt

Práce se zabývá návrhem a vytvořením robotické ruky za použití standardních RC servomotorů, možnostmi jejího ovládání a následným vytvořením programu určeného pro její ovládání. Řeší nejvhodnější způsoby řízení daných servomotorů, zejména s ohledem na jejich co nejjemnější chod.

## Abstract

The thesis deals with the design and creation of a robotic hand using standard RC servomotors, ways of its control and subsequent creation of a program designed for its control. It attempts to solve the most suitable ways of controlling the servomotors, especially with regard to their finest operation.

## Klíčová slova

Robotika, ruka, programování, raspberry pi, řízení, RC servo, ARM, python.

## Keywords

Robotics, hand, programming, Raspberry pi, control, RC servo, ARM, python.

## Citace

DOLEŽAL, Filip. *Robotická ruka s modelářskými servy*. Brno, 2018. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Prof. Dr. Ing. Pavel Zemčík

# Robotická ruka s modelářskými servy

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana profesora Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Filip Doležal  
16. května 2018

## Poděkování

Tímto děkuji za veškerou pomoc při vypracování bakalářské práce svému vedoucímu práce panu Prof.Dr.Ing. Pavlovi Zemčíkovi.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Pohonné jednotky</b>	<b>4</b>
2.1	Servomotory . . . . .	4
2.2	RC servomotory . . . . .	5
2.3	Řízení RC servomotorů . . . . .	7
<b>3</b>	<b>Obvykle používané platformy</b>	<b>9</b>
3.1	Vývojové kity . . . . .	9
3.2	Často využívané DMA knihovny . . . . .	12
3.3	Obvykle používané grafické knihovny . . . . .	14
3.4	Grafické knihovny používané na Raspberry Pi . . . . .	15
<b>4</b>	<b>Existující řešení</b>	<b>17</b>
4.1	Existující hardwarová řešení . . . . .	17
4.2	Existující softwarová řešení . . . . .	20
<b>5</b>	<b>Cíle práce</b>	<b>23</b>
5.1	Návrh robotické ruky . . . . .	23
5.2	Sestavení robotické ruky . . . . .	29
5.3	Prvky návrhu . . . . .	31
5.4	Zapojení servomotorů . . . . .	32
5.5	Výsledný vzhled robotického ramene . . . . .	33
<b>6</b>	<b>Software</b>	<b>34</b>
6.1	Grafický návrh . . . . .	34
6.2	Grafické rozhraní . . . . .	35
6.3	Struktura kódu . . . . .	37
6.4	Funkce software . . . . .	37
<b>7</b>	<b>Testování</b>	<b>41</b>
7.1	Testování vytvořené robotické ruky . . . . .	41
7.2	Testování GUI . . . . .	42
<b>8</b>	<b>Závěr</b>	<b>45</b>
	<b>Literatura</b>	<b>47</b>
<b>A</b>	<b>Návod na sestavení</b>	<b>50</b>

# Kapitola 1

## Úvod

Cílem této práce je vytvořit robotickou ruku za použití komponent, které jsou volně dostupné na českém trhu, a následně jejího řídicího software za použití modelářských RC servomotorů. Práce má za cíl vytvořit dostupnou a konkurenceschopnou variantu již vytvořených produktů. Výstupem by tedy měl být návod pro složení dané robotické ruky i se seznamem jednotlivých komponent a zároveň i samotný software pro její ovládání. Software by měl umožňovat řídit veškeré pohyby dané robotické ruky co možná nejjednodušším způsobem. Toto téma jsem si vybral především z důvodu, že jsem si chtěl vyzkoušet, co takový návrh a programování robotické ruky obnáší. Z technického hlediska mě především zajímaly úskalí konstrukce používající servomotory tak, že tvoří jednotlivé klouby. Z hlediska software jsem pak byl především zvědavý, jak obtížné je jednotlivé klouby donutit fungovat tak, jak by měly.

Robotická ruka je zařízení sloužící k usnadnění a jisté automatizaci práce. Za robotickou ruku můžeme označit v podstatě jakékoliv zařízení sloužící k pohybování předměty. Nejčastěji se s tímto produktem můžeme setkat v automobilovém průmyslu, kde jsou robotické ruce využívány pro svařování, lakování a jiné práce, které vyžadují přesnost. Výhodou robotických rukou je možnost programování jejich pohybu a tedy možnost tento pohyb opakovat se stejnou přesností stále dokola.

Nejedná se však pouze o automobilový průmysl. Robotické ruce se používají ve všech odvětvích, kde je nutná přesnost práce. Výhodou robotických rukou je také možnost práce 24 hodin denně 7 dní v týdnu v podmínkách, ve kterých není možná lidská práce. Obvyklým jevem je tak použití robotických rukou pro přesouvání již hotových výrobků do krabic, skládání krabic a podobné.

Technicky a konstrukčně jsou tyto robotické ruce sice řešeny odlišně od modelářských robotických rukou, ovšem principiálně se jedná o podobná zařízení. U komerčních industriálních robotických rukou je obvyklé, že jsou jednotlivé klouby poháněny motory, které ovšem nejsou přímo v jednotlivých kloubech, ale v základně. Pohon je pak řešen řemeny. Díky tomu je ušetřena váha a ruka tak může zvedat větší břemena. Jak již bylo řečeno, výhody robotických rukou jsou zejména opakovatelnost pohybu, síla a spolehlivost. Prvním kritériem je opakovatelnost pohybu. Ta je důležitá zejména při nutnosti provádět práci stále se stejnou kvalitou nehlédě na čas. Síla robotické ruky je obvykle dána zejména konstrukčními vlastnostmi dané ruky, je ovšem možné říct, že jsou robotické ruce dělitelné dle síly od modelářských až po industriální, které jsou schopny zvedat velice těžká břemena.

Záměrem práce je tedy vytvořit robotickou ruku s pomocí modelářských RC servomotorů tak, aby splňovala určitá kritéria. Robotická ruka by měla být schopná unést břemena s vahou přesahující 100 gramů. Měla by být schopná co nejvolnějšího pohybu, a to tak,

aby bylo dosaženo co možná nejnižšího třepotu samotných servomotorů. Cílem práce je také pokud možno sloučit driver robotické ruky společně s ovládáním robotické ruky do jednoho celku tak, aby nebylo nutné použití dalšího počítače pro ovládání. Cílem je také co možná nejsnadnější replikace, tedy možnost podle návodu v příloze tuto robotickou ruku sestavit a provozovat společně s vyvinutým software. První část je část teoretická, rozebírající jednotlivá úskalí návrhu takového projektu, jednotlivé možnosti řízení jednotlivých servomotorů a možnost použití jednotlivých hardwarových a softwarových prvků. Nejedná se však o encyklopedický přehled všech možností řízení nebo návrhu jednotlivých prvků, vzhledem k zamýšlenému rozsahu práce.

První kapitola popisuje, co jsou to vlastně servomotory a jaká je jejich konstrukce. Druhá kapitola pak popisuje principy, které pro řízení servomotorů existují a jaké problémy s sebou jednotlivé způsoby přinášejí. Tato kapitola také rozebírá pojmem pulsně šířková modulace, o co se jedná a jak se vlastně používá. Následně se tato kapitola věnuje výčtu možných řídicích obvodů, jejich výhodám i nevýhodám. Společně s popisem řídicích desek jsou zde také popsány existující knihovny pro řízení servomotorů i pro vytváření grafických rozhraní. Druhým celkem je pak praktická část návrhu robotické ruky. V první části této práce se seznámíme s výběrem jednotlivých komponent pro části robotické ruky. Další částí je pak popis jednotlivých prvků a částí této ruky. Následuje popis, jak robotickou ruku sestavit, a také se zde nachází detailní popis návrhu ovládajícího software. V závěru této práce jsou pak rozebrány vlastnosti vyvinuté ruky vzhledem k použitým servomotorům, jaké má ruka s těmito servomotory problémy nebo naopak výhody. Budou vyhodnoceny výsledky, kterých se podařilo dosáhnout. Následně budou také rozebrána možná vylepšení, která by bylo možné implementovat jak do ovládacího programu samotné robotické ruky, tak i vylepšení týkající se hlavního hardware robotické ruky.

## Kapitola 2

# Pohonné jednotky

Pro pohon jednotlivých kloubů jsou využívány motory. Obvykle se využívají motory elektrické, ale v industriálních aplikacích není problém se setkat i s motory hydraulickými. Existují v zásadě dva typy motorů. Prvním z nich jsou klasické DC a AC motory, u kterých nezávisí na úhlu jejich otočení a jediné, co se udává jsou, jejich otáčky za minutu, případně otáčky na volt (KV). Druhou kategorií jsou pak motory, používané pro precizní posuny, kdy musí řídicí jednotka daného zařízení přesně vědět, o kolik se pootočila osa motoru. Těmito motory jsou pak servomotory a krokové motory. Tato kapitola však vzhledem k rozsahu této bakalářské práce nemá za cíl fungovat jako encyklopedický přehled všech druhů motorů, jsou tak popsány pouze relevantní druhy.

### 2.1 Servomotory

Servomotor je motor určený pro pohon, existují elektrické servomotory, hydraulické i pneumatické servomotory. Na rozdíl od klasické konstrukce motoru se jedná o zařízení, kterému je možné nastavit přesnou polohu natočení hřídele, obvykle pomocí řídicího signálu.

Díky těmto vlastnostem se tyto servomotory, zkráceně serva, používají pro polohovací aplikace. U těchto servomotorů se dá velice přesně řídit nejen poloha natočení, ale i moment nebo rychlost samotného otáčení. Výhodou servomotorů je také fakt, že jsou schopné dodat vysoký kroutící moment již od nízkých otáček, tedy že mají velice plochou křivku výkonu v závislosti na otáčkách, na rozdíl od konvenčních elektromotorů. [17]



Obrázek 2.1: Servomotor [<https://bit.ly/2IBTldD>]



Servomotory a krokové motory jsou často zaměňovány. Zatímco servomotor obvykle obsahuje klasický DC motor, logiku a potenciometr, pomocí kterého se určuje výsledná poloha, krokový motor používá jiného způsobu dosažení stejného výsledku. Krokový motor využívá více elektromagnetů, kdy jsou tyto elektromagnety ovládány separátně. Je tak možné krokové motory otáčet po jednotlivých krocích, kdy jeden krok znamená posun o předem známý počet stupňů. [15]



Obrázek 2.2: Krokový motor [<https://bit.ly/2lISx9o>]

Tato bakalářská práce se však bude zabývat pouze servomotory, které se budou využívat pro aplikaci v robotické ruce. Jedná se tak především o modelářské servomotory, které mají mírně odlišnou konstrukci od klasického servomotoru. Vzhledem k zamýšlené délce bakalářské práce se tak nejedná o encyklopedický výčet všech typů a druhů servomotorů, které existují společně s jejich ovládáním.

## 2.2 RC servomotory

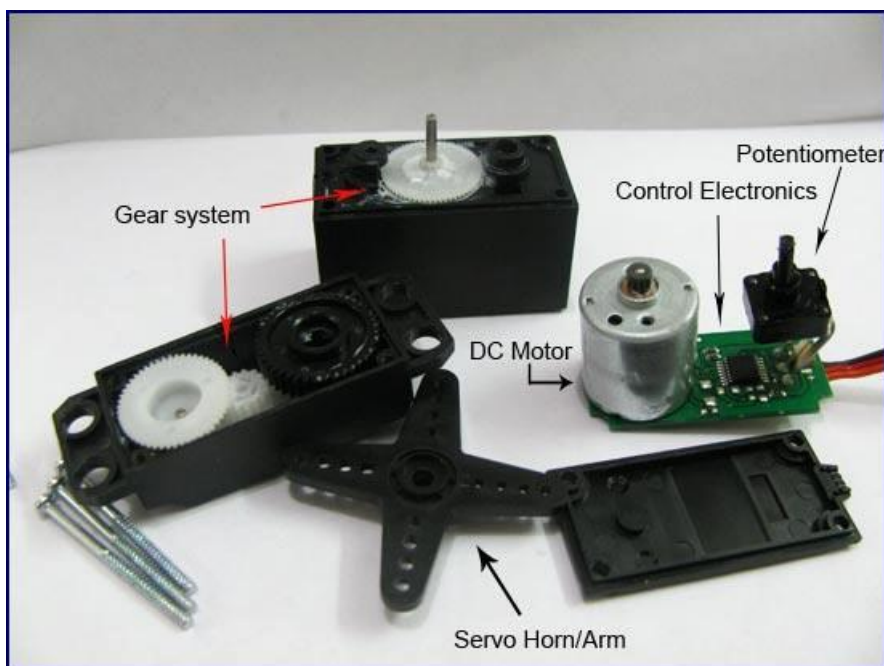
RC servomotory byly primárně určeny pro řízení dálkově řízených rádiových modelů, proto se těmto servomotorům také někdy říká modelářské servomotory. Základním principem takového servomotoru je na základě vstupního signálu pootočit svou výstupní hřídel do správného úhlu. [19] Díky této vlastnosti se však velmi hodí i pro použití v robotice, kde takovéto modelářské servo v sobě integruje hned několik dílů nutných pro přesný pohyb ramen. Místo použití klasických servomotorů, optických bran a senzorů konečných poloh pro určení přesné polohy daného servomotoru [8] za nás všechnu tuto práci obstará elektronika daného RC servomotoru. [4]

### 2.2.1 Konstrukce RC servomotoru

Konstrukce RC servomotorů je napříč nabízenými servomotory vždy v podstatě stejná. Modelářské servo se skládá ze tří základních částí. První částí je stejnosměrný motor, který zajišťuje pohyb výstupní hřídele, následně servomotor obsahuje převodovku, která je schopná převést vysoké otáčky stejnosměrného motoru na nízké otáčky, ale zato s mnohem větším kroutícím momentem. Druhou funkcí převodovky v RC servomotoru je zajištění přenosu

na zpětnovazební potenciometr, který následně říká řídicí elektronice, zda byl již dosažen správný úhel výstupní hřídele, a je tak nutné zastavit další otáčení a nebo se má stále pokračovat. Poslední částí RC servomotoru je samotná elektronika.[8]

Řídicí elektronika má za úkol zpracovat vstupní PWM signál, na základě kterého vyhodnocuje stupeň natočení. To je porovnáváno s natočením zpětnovazebního potenciometru. Obvyklé jsou pulsy o délce 1 ms až 2 ms pro úhly natočení od 0 do 180 stupňů, kdy se servo plynule natáčí požadovaným směrem. Minimální hodnota natočení je u lepších servomotorů 1 stupeň, ovšem obvyklejší hodnota jsou 2 stupně.[29]



Obrázek 2.3: Rozbor servomotoru [<https://bit.ly/2rEHLoj>]

### 2.2.2 Typy RC servomotorů

V současnosti se používají zejména rotační servomotory, kdy funkce spočívá v natočení výstupní hřídele podle vstupního signálu. V dnešní době existuje mnoho typů těchto servomotorů, zejména však s úhlem natočení 120 nebo 180 stupňů. Zároveň existují také lineární servomotory, které jsou založeny na převodu rotačního pohybu na pohyb posuvný, jinak je jejich řízení totožné. Tento typ se používá zejména při použití u modelů letadel pro funkce vysunutí a zasunutí podvozku do těla samotného letadla. Existují také RC servomotory označované za kontinuální, nebo někdy také jako 360stupňová serva, kdy servomotor nemá žádné krajní polohy, a je tak možné je využít třeba pro navíjení lanoví u modelů lodí.

Jako druhá možnost pro dělení RC servomotorů se nabízí dělení na základě vestavěné řídicí elektroniky, a to na analogová a digitální. Z hlediska řízení těchto servomotorů není tento rozdíl podstatný, jelikož se oba typy řídí stejným PWM signálem. Digitální servomotory jsou však přesnější a silnější, ovšem za cenu většího odběru proudu. [26]

## 2.3 Řízení RC servomotorů

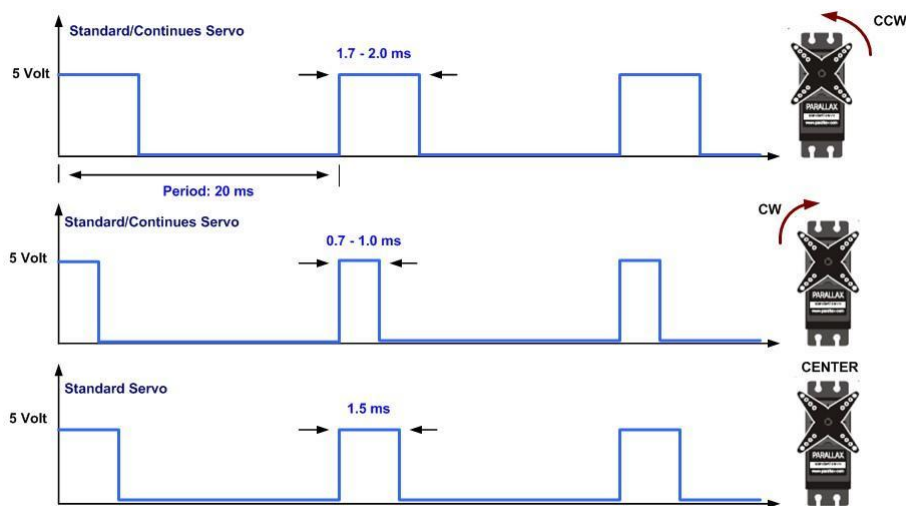
Nejčastějším typem řízení servomotorů je pomocí PWM signálu, jedná se o sérii opakujících se impulsů proměnné délky. Druhou možností je měnit pracovní cyklus. Samotné servo má standardně 3 vývody:

- Červený +5V
- Černý GND
- Bílý/oranžový – signálový vodič

Moderní RC servomotory svoji pozici mění pouze na základě šířky pulzu a měnit „duty cycle“ není potřeba. Většina dnešních servomotorů se přemístí do své středové polohy (90 stupňů u 180stupňového servomotoru) při délce pulsu 1,5 ms nehledě na to, jak často tento puls dostávají. [29]

Jako standardní perioda se dnes udává 20 ms nebo taky 50 Hz, jedná se o ekvivalentní způsoby zápisu periody. Za předpokladu, že servomotor dostane na svůj vstup puls nižší než 1,5 ms, pak se servo začne otáčet proti směru hodinových ručiček (o kolik stupňů záleží na délce pulsu). Za předpokladu, že je signál delší než 1,5 ms, pak se začne servomotor otáčet po směru hodinových ručiček. Každý servomotor však může mít tyto hodnoty jiné, a tak je vždy nutné zkontrolovat jednotlivé hodnoty s datasheetem daného servomotoru. Servomotory jsou schopné udržovat svoji pozici proti vynaložené síle, která se je snaží vychýlit až do maximální síly servomotoru udávané v datasheetu. Toto chování však nastává pouze v případě opakování daného pulsu každých 20 ms.

Pro řízení servomotorů je tak nutné používat PWM signály, ovšem záleží, jakým způsobem tyto signály vyrábíme. Pokud je zvolen špatný typ generování může se stát, že nebude PWM signál pevný a servomotor bude oscilovat.



Obrázek 2.4: PWM řízení servomotoru [<https://bit.ly/2L06hZt>]

### 2.3.1 Vytváření pulsně šířkové modulace (PWM)

Pro vytváření PWM pulsů existují vesměs dva možné přístupy, klasické vytváření impulsů pomocí vestavěných funkcí jednotlivých mikrokontrolérů či procesorů. Jedná se o nejjednodušší možnost, jak dosáhnout analogového chování výstupního pinu pomocí digitálního přístupu. [12]

Většina takto vyprodukovaných signálů je vytvořena pomocí určité variace funkce *analogWrite()*, takto vyprodukovaný PWM signál je zcela dostačující pro většinu aplikací, ať už se jedná o ztlumování svitu LED nebo řízení otáček elektromotoru. Dostačující jsou proto, že nedokážeme vnímat jemné odchylky v šířkách těchto pulsů. [10]

Tyto chyby vznikají díky tomu, že vytváření těchto pulsů řídí procesor. Prvním problémem je určitá zátěž procesoru, kterou ale budeme v tomto případě ignorovat. Mnohem větším problémem je, že procesor musí obsluhovat také svoje vlastní přerušení, které mají už z principu vždy prioritu. V případě, že přerušení přijde přesně ve špatný čas, se puls buď zkrátí a nebo naopak prodlouží, a to ve chvíli, kdy je řízen servomotor a jedná se o pohyb i o 2 stupně způsobuje chvění samotného servomotoru, kdy servomotor osciluje mezi různými pozicemi.

### 2.3.2 Direct memory access (DMA)

Tento problém řeší metoda přímého přístupu do paměti. Jedná se o metodu, kdy je z rovnice odstraněn procesor a data jsou přenášena přímo mezi pamětí a vstupně výstupním zařízením. [16] Díky tomuto chování můžeme v paměti vytvořit masky, které se následně na každém PWM kanále postarají o příslušnou šířku pulsu, to znamená, že se tyto masky přímo přiřadí/zapíší do registrů jednotlivých portů.

Díky tomuto chování je možné přesně specifikovat šířku pulsů, aniž bychom museli počítat s problémy, které způsobí samotný procesor nebo operační systém. Zbavíme se tak klepání servomotorů, tedy tzv. jitteru. Jediný problém tak při tomto přístupu může nastat ve zpožděném zápisu nové hodnoty pro šířku pulsu, tedy může být začátek pohybu ruky mírně zpožděn. Ovšem toto zpoždění je v podstatě zanedbatelné. [28]

## Kapitola 3

# Obvykle používané platformy

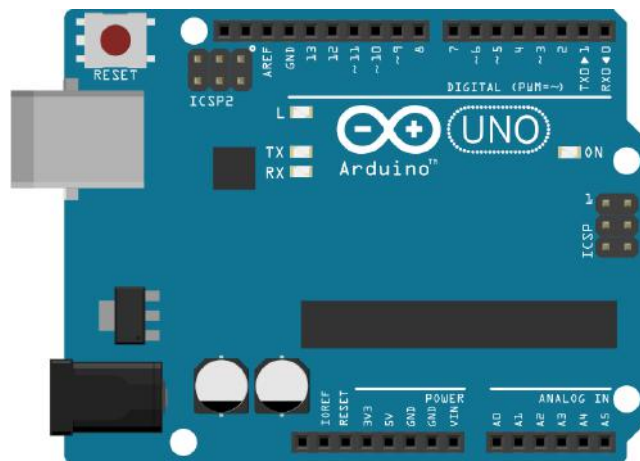
Platformem, které mohou řídit robotické prvky, je celá řada. Hlavním kritériem, které udává použitelnost dané platformy pro robotiku, je v zásadě pouze její míra programovatelnosti. Druhým kritériem je pak počet výstupních pinů, které se dají použít pro ovládání jednotlivých mechanismů. Takovýchto platformem, nebo také vývojových desek, které se na trhu objevují, je celá řada, tato práce však nemá za cíl rozebrat všechny existující platformy. Nejedná se tak o encyklopedický přehled, ovšem pouze o nástin současné situace.

### 3.1 Vývojové kity

Při výběru platformy, pomocí které bude celá ruka řízena, bude kladen důraz na několik málo kritérií, které ale každá platforma musí splňovat tak aby se dala považovat za použitelnou pro daný projekt:

- Podpora DMA
- Výsledná jednoduchost řešení
- Existence knihoven

Prvním kritériem tak je najít platformu, která by podporovala použití DMA pro vytváření PWM signálů. Jako první možností bylo uvažováno Arduino. Problémem Arduina jakékoliv verze však je, že ačkoli podporuje DMA přenosy tak pro řízení PWM již toto chování není možné použít. Druhým problémem řešení, které by využívalo Arduino desku, by byla nutnost specifikovat komunikační protokol mezi samotným Arduinem a počítačem, na kterém by běžela aplikace. Tedy prvotní myšlenka zařízení, které by dokázalo jak obsluhovat robotickou ruku, tak se chovat i jako samotný počítač, tato varianta nesplňuje. Muselo by se jednat o Klient server aplikaci, která ovšem není cílem tohoto projektu. [\[12\]](#)



Obrázek 3.1: Arduino UNO [<https://bit.ly/2GdmfLS>]

Dalším možným řešením je použití nějakého široce dostupného ARM procesoru, ovšem toto řešení má také několik nevýhod, které se ukázaly jako fatální. Prvním problémem je samotný návrh takového řešení, kdybychom buď museli navrhnout celý podpůrný obvod pro ARM procesor, kdy by pak replikace tohoto řešení uživatelem, který nechce osazovat desku a pájet SMD komponenty, byla zcela nereálná. Druhou variantou je použití jednoho z mnoha vývojových kitů takovýchto ARM procesorů.[3] Opět ale narážíme na problém, kdy by tato deska byla pouze prostředníkem, nebo přesněji driverem, mezi samotnou rukou a počítačem. Tedy bychom se opět nevyhnuli řešení a návrhu komunikačního kanálu mezi těmito zařízeními.



Obrázek 3.2: Atmel ARM development Kit [<https://bit.ly/2L0ANSQ>]

Nutné tedy bude najít desky, které tyto parametry splňují, tedy kombinují funkce klasického počítače i funkce řízení výstupních pinů. Těchto desek na trhu existuje nepřeberné množství od desek využívajících operační systém Windows až po systém Linux. V rámci otevřenosti dané platformy však budou brány v potaz pouze desky, na kterých běží operační systém Linux. [23]



### 3.1.1 Raspberry Pi

Dalším kandidátem je použití široce rozšířeného řešení v podobě Raspberry Pi, které díky obrovské skupině vývojářů splňuje všechny vstupní požadavky pro jeho použití v tomto projektu. Samotné Raspberry Pi podporuje DMA přenosy, které je možné použít i pro řízení PWM signálů. Zároveň se jedná o plnohodnotný, i když ne vysoce výkonný počítač běžící na platformě Linux s přímo vytvořenou distribucí Raspbian pro tuto platformu. Tento operační systém je upravenou distribucí oblíbeného operačního systému Debian. V neposlední řadě také existuje množství knihoven, které usnadní práci se samotným generováním PWM signálů pomocí DMA. [22]

Samotného Raspberry Pi existuje několik variant. [9] Od v podstatě holých desek, jako je třeba Raspberry Pi Zero až po nejnovější a nejvybavenější Raspberry Pi 3. V rámci toho, že výběr možné verze Raspberry Pi bude následně velice omezovat i možnost použitých knihoven, musíme volit verzi Raspberry Pi zejména podle budoucí používané knihovny, a to buď původní Raspberry Pi v některé jeho verzi, které se lišily zejména ve velikosti RAM mimo jiných menších změn. Druhou možností je využití druhé verze Raspberry Pi. Díky tomu, že na tyto verze Raspberry Pi existuje velice dobrá knihovna pro DMA PWM, která ale bohužel není kompatibilní s poslední verzí daného počítače, bude vytváření přesných PWM signálů pomocí DMA vytvářeno pomocí této knihovny. [21]

Zejména díky výkonovému propadu u Raspberry Pi první generace oproti verzi 2 je vhodné volit minimálně verzi Raspberry Pi 2, která disponuje těmito:

- Procesor SoC BCM2836 z rodiny ARM Cortex-A7 obsahující 4 jádra na frekvenci 900Mhz
- 1GiB RAM (+50
- GPU VideoCore IV podporující OpenGL ES 2.0 a rozlišení 1080p při 30FPS
- Slot microSD, 4x USB2.0, RJ45 10/100
- GPIO vývody

Díky vlastnostem samotného rozložení desky, kdy napříč všemi majoritními verzemi zůstává zachováno rozložení GPIO pinů není problém nahradit verzi 2 za novější verzi 3.



Obrázek 3.3: Raspberry Pi 2 mod. B [https://bit.ly/1m1YfSJ]

## 3.2 Často využívané DMA knihovny

Pro DMA přenosy, které umožňují vytváření PWM pulsů, existují často využívané knihovny, které usnadňují práci při programování samotného ovládacího programu pro řízení servomotorů pomocí generovaného signálu, případně celé robotické ruky. Jedná se tedy zejména o knihovny, které usnadňují práci s DMA přenosy při generování PWM signálu, a knihovny usnadňující práci při vytváření grafického prostředí samotné aplikace.

Základním požadavkem je pak hledání takové knihovny pro generování PWM pomocí DMA, která je a bude schopná obsluhovat minimálně daných 6 servomotorů. Je totiž obvyklé, že robotické ruce využívají minimálně 6 os pohybu, tedy že využívají alespoň daných 6 servomotorů při pohybu. [25] Vzhledem k tomuto požadavku na funkčnost této knihovny jsme však prakticky omezeni pouze na dvě použitelné knihovny, a to RPIO a ServoBlaster. Jedná se však o dvě nejznámější a zároveň nejrozšířenější knihovny pro DMA řízení. Výhodou vysoké oblíbenosti těchto knihoven je, že existují porty od různých vývojářů pro nejnovější verze desek.

### 3.2.1 ServoBlaster

Nejedná se ani tak o knihovnu, jako spíš o software poskytující interface pro řízení jednotlivých servomotorů připojených na jednotlivých GPIO pinech. Jedná se vlastně o sadu příkazů, pomocí kterých se dá ovládat pozice servomotoru.

V případě, že by byl ovládací software psán v Bashi nebo podobném skriptovacím jazyce tak by tento software byl velmi použitelný. Jelikož po provedení příkazu je signál udržován až do doby, než se zadá nový příkaz. Primárně je tento software určen pro řízení 8 servomotorů, ovšem dá se nakonfigurovat až pro řízení 21 servomotorů. Knihovna nebo software pracuje s přesností 10 us, což je dostatečné.[18]

Ačkoli je tento software velice propracovaný, tak pro využití při programování ovládacího software se nehodí. Což ovšem neznamená, že se jedná o nevyužitelný software. Jedná se o perfektní software pro testování pohybů jednotlivých ramen, jejich maximálních a minimálních pozic a také zjištění, zda bude muset být servo trvale napájeno „tedy trvale mu bude muset být posílán signál“, aby nedocházelo ke klesání samotného ramena ruky, a nebo nebude toto chování potřeba.

Použití tohoto software vypadá tak, že samotná instalace driveru vytvoří soubor `/dev/servoblaster`, do kterého jsou posílány všechny příkazy:

- Formát příkazu
  - `<servo-number>=<servo-position>`
  - `P<header>-<pin>=<servo-position>`
- Ukázka vygenerování signálu o šířce 1,2 ms na servomotoru 3
  - `echo 3=120 > /dev/servoblaster`
  - `echo P1-13=120 > /dev/servoblaster`
- Ukázka mapování servomotorů více viz github



Servo number	GPIO number	Pin in P1 header
0	4	P1-7
1	17	P1-11
2	18	P1-12
3	21/27	P1-13

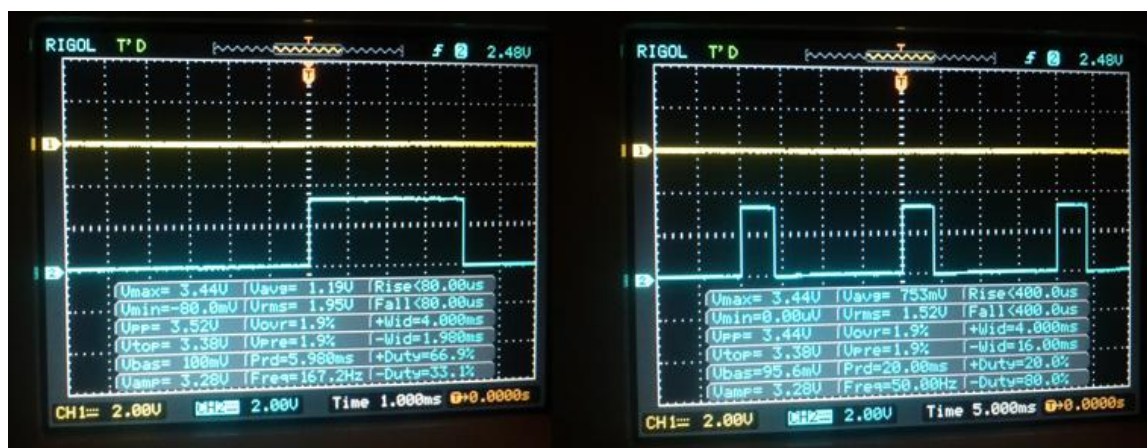
Tabulka 3.1: Pin mapping v knihovně ServoBlaster

### 3.2.2 RPIO.PWM

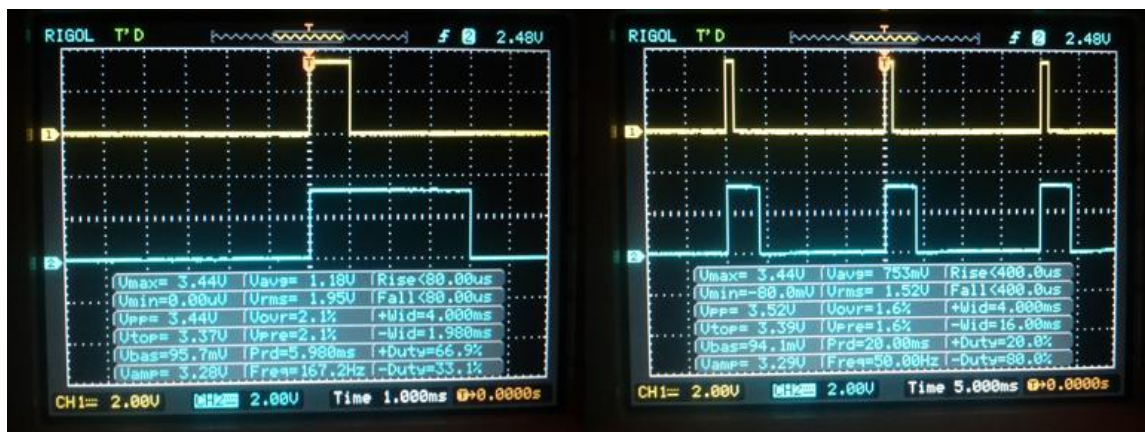
Jedná se o knihovnu speciálně napsanou pro Raspberry Pi. Poskytuje rozhraní pro řízení PWM signálu pomocí DMA přenosů s maximálním rozlišením 1 us. Poskytuje možnost řídit jakýkoliv z 15 DMA kanálů a jakékoliv množství GPIO na kanál, nejsme tedy nijak omezeni, co se týče konečného počtu servomotorů, kromě počtu GPIO pinů.

Díky tomu, že je PWM signál generován pomocí DMA přenosů a ne pomocí klasických metod generování, nepoužívá tato knihovna téměř žádné prostředky CPU, díky čemuž je schopná generovat velice přesné impulsy. Tyto vlastnosti pak zaručují možnost zbavení se jitteru samotných servomotorů. [20]

Samotná knihovna je napsána v jazyce C, je tedy použitelná v jakémkoliv programu napsaném v tomto jazyce. Ovšem pro potřeby této práce bude potřeba wrapper nad touto knihovnou, který nám umožní ji používat v jazyce Python. Knihovna disponuje dvěma základními přístupy. Prvním je low level přístup, kdy je možné veškeré aspekty signálu kontrolovat ručně. Druhou možností jsou již předpřipravené pomocné funkce, které usnadňují práci se samotnými servomotory. Při nastavování jednotlivých pulsů je stejně jako u ServoBlaster použita jako minimální hodnota 10 us, ovšem při low-level přístupu je možné zvýšit citlivost až na 1 us. Taková citlivost je však pro zamýšlené použití zbytečná, jelikož puls 1us obvykle znamená přesnost 0,1, které klasické analogové servomotory stejně nejsou schopné zobrazit. Obvyklá citlivost klasického RC servomotoru je 1. Níže jsou na osciloskopu vidět výstupy jednotlivých pulsů, kdy je vidět, že pulsy jsou pokaždé stejné. Na druhém obrázku je vidět možnost přidání dalších pulsů do jednotlivých subcyklů. [21]



Obrázek 3.4: Generované signály pomocí RPIO.PWM knihovny [https://bit.ly/2jXJzom]



Obrázek 3.5: Další generované signály pomocí RPIO.PWM knihovny [https://bit.ly/2IGgUC4]

Knihovna, popřípadě její wrapper, je v její původní verzi určena pouze pro práci s Raspberry Pi první generace. Toto omezení však znamená, že je knihovna nepoužitelná s novějšími iteracemi Raspberry Pi. Je tedy nevyužitelná pro aplikace, kdy musí být počítač dostatečně výkonný nejen pro řízení samotných servomotorů, ale také na rychlé zobrazování GUI samotného operačního systému daného Raspberry Pi.

Tento problém je však již vyřešen, v současné době existují porty této knihovny pro druhou i třetí, tedy poslední, verzi Raspberry Pi. Toto přepsání sice znamenalo úpravu podstatné části původního kódu vzhledem k rozdílné práci s pamětí a také změně paměťových map, ale díky tomu je zajištěn snadný port na nové verze raspberry Pi. Je však nutné mít na paměti, že se již nejedná o původní verzi dané knihovny, ale o její porty. Je tak možné narazit na jisté problémy, které se nemusely vyskytovat v originální verzi knihovny. [20]

### 3.3 Obvykle používané grafické knihovny

Grafická knihovna je knihovna programu specializující se na renderování počítačové grafiky na display. Typicky se tak jedná o soubor optimalizovaných funkcí pro renderování jednotlivých prvků. Grafické knihovny pak mohou fungovat ve dvou základních módech, kdy první z těchto módů je renderování čistě pomocí software a pomocí procesoru. Druhou možností je pak využití hardwarové akcelerace pomocí grafické karty, kdy hlavním úkolem této grafické knihovny je sestavit obraz a vykreslit jej na obrazovku. [27]

Grafické knihovny je pak možno dělit na dvě kategorie. Těmito kategoriemi jsou knihovny pro práci s 3D grafikou a 2D grafikou. Knihovny pracující s 3D grafikou jsou používány zejména pro tvoření počítačových her a simulací.

Pro účely vytváření grafických rozhraní k programům, které ovládají nějaké zařízení nebo poskytují rozhraní pro jiné účely, se však 3D grafika nehodí. Je tak nutné využití 2D grafických knihoven. Obvykle je pro jejich multiplatformitu volen jazyk C++. [27]

Nejrozsáhlejší grafickou knihovnou, která podporuje většinu platform, je pak QT knihovna. Tato knihovna je široce užívána pro vytváření aplikačního software a grafická uživatelská rozhraní. Jedná se o grafickou knihovnu napsanou v C++ a primárně určenou pro C++, ovšem existují i verze pro Python (PyQt), Ruby (QtRuby), C, Perl, Pascal a jiné. Aplikace napsané pomocí této knihovny je pak možné distribuovat pod GPL licenci.

OS	Vulkan	Direct X	GNMX	Metal
Windows 10	Free, Nvidia and AMD	Free, MS	no	no
Mac	MoltenVK	no	no	Free, Apple
GNU/Linux	Free	no	no	no
Android	Free	no	no	no
iOS	MoltenVK	no	no	Free, Apple

Tabulka 3.2: Grafické knihovny pracující s 3D grafikou

OS	GTK	QT X	WxWidgets	FLTK	FOX
Windows 10	Yes	Yes	Yes	Yes	Yes
Mac	Yes	Yes	Yes	Yes	Yes
GNU/Linux	Yes	Yes	Yes	Yes	Yes
Android	No	Yes	No	No	No
iOS	No	Yes	No	No	No

Tabulka 3.3: Grafické knihovny pracující s 2D grafikou

V případě komerčního užití je nutné zakoupení licence. [14] Grafických knihoven, které jsou však použitelné na Raspberry Pi, existuje také celá řada. Některé se pro vytváření složitějších grafických rozhraní hodí více, některé méně. Existují také knihovny, které se přímo specializují na práci s dotykovými obrazovkami. Dále je výčet nejvyužívanějších z nich. [24]

## 3.4 Grafické knihovny používané na Raspberry Pi

### 3.4.1 Kivy

Jedná se o knihovnu uzpůsobenou přímo pro práci s dotykovými obrazovkami. Celá knihovna funguje nad frameworkem OpenGL ES 2. Tato knihovna je událostmi řízená, kdy je celý framework založen na architektuře hlavního main cyklu. Knihovna se tak hodí zejména pro návrh grafických her.

Je však použitelná i pro kreslení GUI, kdy je možné jednotlivé akce vyvolávat pomocí triggerů. Je možné vytvářet vidgety ale i animace. [13]

### 3.4.2 Pyfroms

Jedná se o knihovnu specificky napsanou pro python, ať už verze 2.7 nebo 3 a vyšší. Je to knihovna, která podporuje přenositelnost a modularitu kódu s co možná nejmenším úsilím. Je navržena na základě OpenGL a PyQt.

Tato knihovna je určena zejména pro tvoření software založeného na formulářovém rozložení s mnoha tlačítky a vstupními prvky. Je to tak jedna z možností, kterou je možno použít při tvorbě GUI pro software na ovládání ruky. [13]

### 3.4.3 PyQT

PyQt je multiplatformní knihovna, psaná v C++. Výhodou této knihovny je její masivní rozšířenost. Je tak snadné i pro naprostého začátečníka začít s ní programovat. Výhodou je také velké množství nástrojů, které se dají s touto knihovnou použít, jako je třeba Qt Creator, který usnadňuje práci při prvotním návrhu GUI.

Jedná se o open-source knihovnu, ale pouze v případě, že je software s ní vyvíjený poskytnut na základě otevřené licence. V opačném případě je nutné zakoupit si licenci. Pro využití v této práci se hodí zejména pro svou jednoduchou implementaci do pythonu a také díky množství návodů, které je možno využít při práci s ní. [13]

### 3.4.4 PyGUI

Poslední popisovanou knihovnou je knihovna PyGUI, je to knihovna, která je oproti Kivi nebo PyQt skutečně jednoduchá. Zato je ale velmi intuitivní a snaží se mezi GUI a python kód nevkládat zbytečné sekvence kódu. Její myšlenkou je spojení GUI a python kódu co možná nejjednodušším způsobem. [13]

## Kapitola 4

# Existující řešení

Na trhu již existují řešení, která jsou podobná této práci. V této kapitole tak budou rozebrány některá existující řešení, která jsou na trhu k dostání. Nejedná se však o kompletní přehled všech stavebnic či řešení, které jsou ve světě dostupná. Dále jsou pak rozebrány zejména rozdílné druhy, na které je možné narazit.

### 4.1 Existující hardwarová řešení

Na trhu tedy existuje podobných řešení mnoho, ať už se jedná o ruce vyrobené pomocí 3D tisku nebo přímo celé stavebnice. Takovéto stavebnice můžeme dělit do dvou základních kategorií na základě použitých materiálů. První kategorií jsou Robotické ruce určené zejména na hraní, kdy je použit zejména plast a otáčení jednotlivých kloubů je řešeno obvykle jiným způsobem, než jsou RC servomotory.

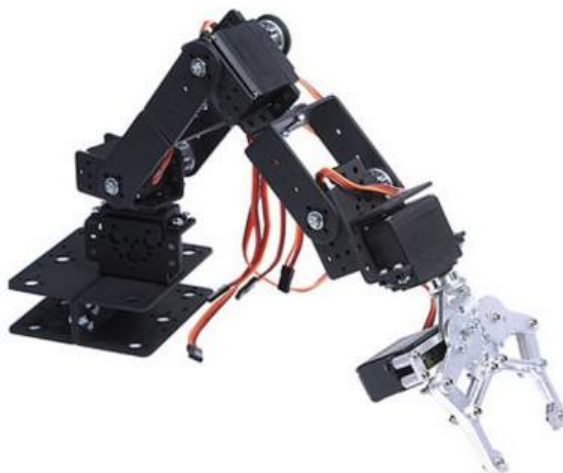
Zástupcem této kategorie může být třeba „Stavebnice Robotické rameno“, kdy je tato stavebnice určena pro děti od 14 let a ovládání je řešeno pomocí klasického ovladače. Ovšem tato robotická ruka je zcela generická a mnoho výrobců ji nabízí pod svou značkou, namátkou třeba jako Cebeekit, OWI-535 nebo jako C9895. [6]



Obrázek 4.1: C9895 robotická ruka [<https://bit.ly/2rGI9SN>]

Druhou kategorií jsou robotické ruce využívající klasické RC servomotory. Ttyto stavebnice se obvykle prodávají ve dvou variantách. První z nich je pouze instalační materiál,

jako jsou brackety, šrouby, hliníkové profily apod. Je tak zcela na uživateli, aby nakoupil a nainstaloval potřebné servomotory a aby vymyslel, jak tyto servomotory ovládat. Občas se tyto verze prodávají společně s potřebnými servomotory, ovšem jejich řízení a programování je opět zcela ponecháno na uživateli. [1]



Obrázek 4.2: Generická stavebnice robotické ruky [<https://bit.ly/2wFU60v>]

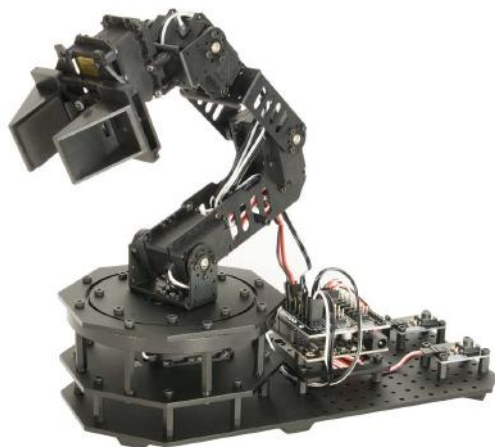
Druhou variantou v této kategorii jsou již hotové stavebnice, které obsahují obvykle kompletní instalační materiál, všechna serva a hlavně i řídicí desku a napájení. Nevýhodou těchto stavebnic je velmi často proprietární ovládací deska, obvykle založená na nějakém Atmega procesoru. Toto řešení pak obvykle nabízí přímo od výrobce možnost tyto procesory programovat, ovšem ve většině případů pouze v proprietárním software od výrobce. (Arexx RA2-Mini) [7]



Obrázek 4.3: Arexx RA2-Mini [<https://bit.ly/2IeNOur>]

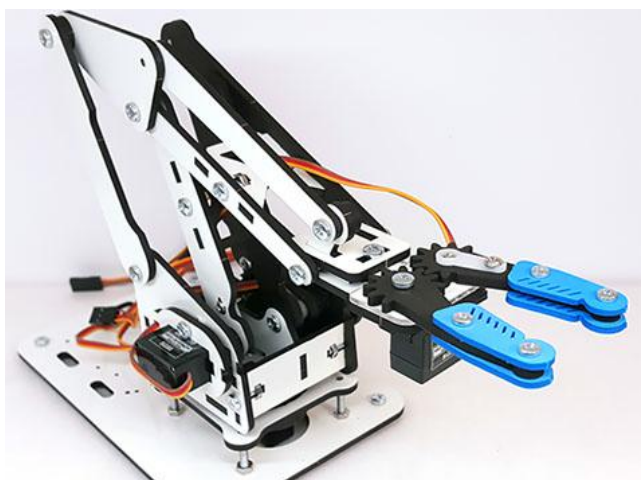


Ve většině případů návody pro výrobu podobných robotických rukou spoléhají na použití nějaké verze arduina, kdy je jeho naprogramování v rukou samotného uživatele. Druhým přístupem při použití arduina je software InterbotiX Arm Link Software, který se primárně soustředí na poskytnutí grafického rozhraní pro ovládání rukou postavených na arduinu. V tuto chvíli je tímto softwarem možno ovládat zejména produkty společnosti RobotGeek. [5]



Obrázek 4.4: RobotGeek Snapper Arm [<https://bit.ly/2lh6KVb>]

Posledním typem robotických rukou jsou stavebnice, které jsou laserově vyřezané a jsou prodávány jako kompletní stavebnice. V tomto případě se jedná o stavebnici ArmUno 2.0, která sestává z jednotlivých částí ruky vyřezaných z plastu. Stavebnice se dodává bez servomotorů a řídicí desky, je tedy nutné tyto díly nakoupit separátně a následně podle návodu robotickou ruku poskládat. Výhodou je, že díky tomu tato stavebnice podporuje široké množství řídicích desek od Arduino Uno až po Raspberry Pi 3. Celá stavebnice je také konstruována spíše jako konstrukce lžice bagru. Nejedná se tedy o servo v kloubu. [2]



Obrázek 4.5: ArmUno 2.0 Robotic Arm [<https://bit.ly/2Ijz9Km>]

## 4.2 Existující softwarová řešení

Softwarová řešení pro řízení existujících stavebnic robotických rukou jsou obvykle použitelné pouze pro danou stavebnici, popřípadě pouze pro stavebnice od daného výrobce. Je možné najít různé typy ovládacího software pro různé projekty. Dále je výčet pouze některých z nich.

### 4.2.1 USB interface for OWI-535

Jedná se o řídicí software určený pro robotickou ruku OWI-535 od firmy OWI robots. Je to ovšem generická ruka, kterou různí výrobci nabízejí pod různými názvy.

Jak je vidět na obrázku, ovládací software nabízí dva módy operace. Prvním z nich jsou tlačítka umístěná u kloubů ruky, kdy každé tlačítko hýbe daným kloubem na každou stranu. Druhým modelem je pak programovací mód kde se robotické ruce předávají příkazy, které následně vykonává. [6]



Obrázek 4.6: OWI-535 ovládání [<https://bit.ly/2rGVjin>]

Druhou variantou tohoto software je podobný ovládací program s rozdílným grafickým rozhraním, ovšem pro ruku C9895 od společnosti Quasar electronic. Jedná se o totožnou robotickou ruku, kdy je ale grafické rozhraní upraveno pro modernější vzhled.

Tato verze pak na rozdíl od předchozí přidává k funkčnosti také ovládání pomocí klávesnice. Opět se setkáváme s dvěma módy operace, ovládání pomocí tlačítek buď v grafickém rozhraní, nebo pomocí tlačítek na klávesnici. Druhou možností je opět ovládání pomocí skriptu, ve kterém se nacházejí jednotlivé kroky pro každý kloub.

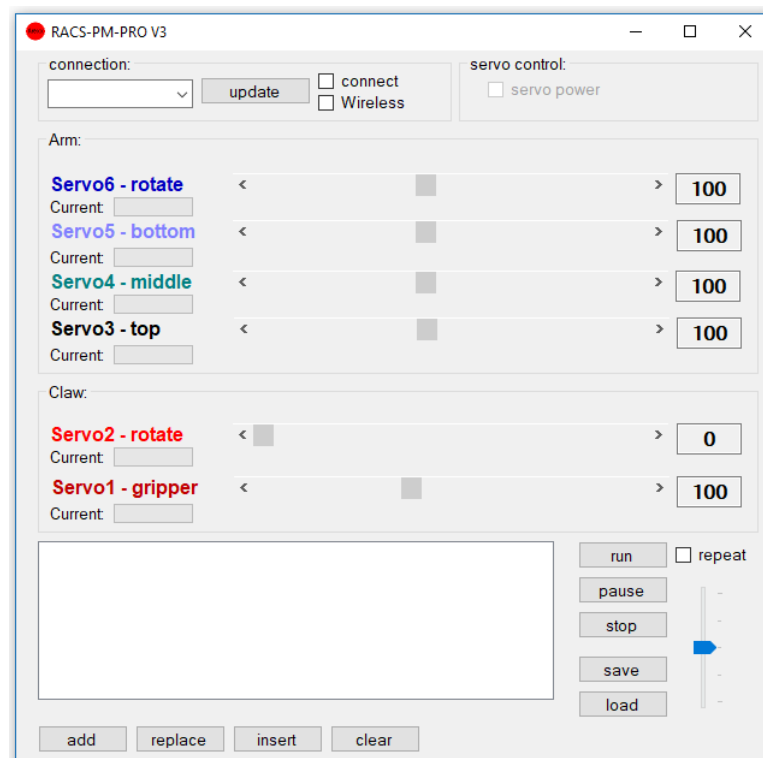




Obrázek 4.7: C9895 ovládací software [<https://bit.ly/2GfRRAu>]

#### 4.2.2 RACS-PM-PRO V3

Tento řídicí software je určený pro robotické ruce od výrobce AREXX engineering. Primárně je tento software určený pro řízení modelu RA1-PRO a RA2-PRO, kdy celá stavebnice opět funguje v modelu klient server. [7]



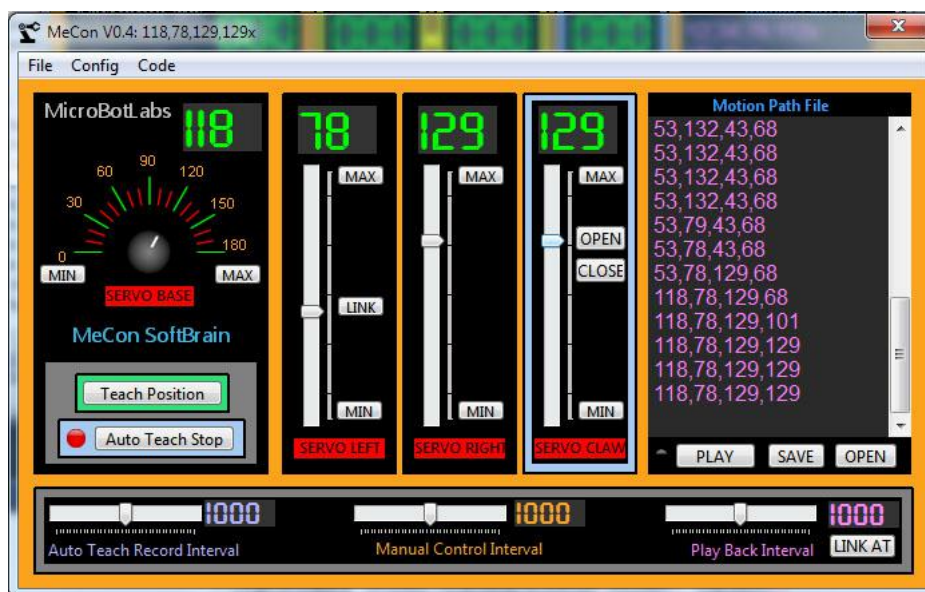
Obrázek 4.8: RACS-PM-PRO V3

Na rozdíl od předchozích variant ovládacího software tento nestaví na líbivém vzhledu, ovšem obsahuje zejména posuvníky pro ovládání jednotlivých servomotorů napřímo.

Tento software také obsahuje dva typy možného ovládání, a tím je opět provádění příkazů nebo hýbání klouby pomocí posuvníků. Na rozdíl od předchozích řídicích software je tento schopný fungovat i v bezdrátovém režimu.

#### 4.2.3 MeCon

MeCon software je určen pro ovládání robotické ruky MeArm a ArmUno, které jsou popsány výše. Hlavní myšlenkou je vytvoření univerzálního software pro různé platformy. Nejedná se však o multiplatformní systém, je to pouze ovladač, který dokáže komunikovat s ovládacím software dané stavebnice, který může běžet na desce od Arduina až po Raspberry Pi. [2]



Obrázek 4.9: MeCon software [<https://bit.ly/2Gerr23>]

Tento software je určený pro operační systém Windows a staví na komunikačním portu COM nebo na USB portu. Software opět obsahuje posuvníky pro ovládání jednotlivých servomotorů, ale i příkazovou část. Tento software umožňuje zachytávání jednotlivých pohybů do takzvaného motion path file, který snímá aktuální polohu všech servomotorů na základě přednastavené frekvence. Při pohybu posuvníku je pak automaticky tato skutečnost uložena do tohoto souboru. Následně je možné takto vytvořený soubor přehrát.

## Kapitola 5

# Cíle práce

Cílem této práce je vytvořit stavebnici robotické ruky složenou zejména z volně dostupných komponent tak, aby po uživateli nebylo požadováno použití různých dalších nástrojů, jako jsou 3D tiskárny a podobné. Výstupem by měl být podrobný návod na sestavení robotické ruky, pro kterou se poté dá použít dodávaný ovládací software.

### 5.1 Návrh robotické ruky

V současné době na trhu existuje mnoho řešení robotických rukou. Jedná se buď o stavebnice určené ke složení, kdy je hlavním cílem rozvíjení technické zdatnosti uživatele nebo jsou to pak kompletní robotické ruce, používající proprietární díly. Jako jistý mezičlánek se dají považovat stavebnice obsahující všechny potřebné díly pro složení funkční robotické ruky, jakými jsou servomotory, brackety a hlavně ovládací deska.

Zásadním problémem těchto řešení je v případě kompletních rukou, jakou je třeba C9895 od společnosti Quasar electronic, použití nestandardních dílů. V případě této robotické ruky jsou dokonce jednotlivé klouby řešeny pomocí standardních DC motorů a jednoduchých převodovek (šnek a ozubené kolo), díky čemuž není možné dosáhnout vysoké přesnosti pohybu ruky, ovšem je zajištěno, již z vlastností takové převodovky, neklesání ramene. [2]

Na druhém konci spektra jsou pak robotické stavebnice, sestavené z klasických servomotorů, ať už digitálních nebo analogových, kdy jsou jednotlivé klouby tvořeny těmito servomotory společně s brackety pro jejich uchycení. Tento přístup je zcela vyhovující. Zásadním problémem však je, že každý výrobce používá, v případě kompletních stavebnic, obvykle řídicí desku sestavenou za použití nějakého mikrokontroleru, kdy je zásah do jejího řídicího softwaru buď nemožný, nebo v některých případech možný pouze pomocí software od výrobce.

Jistým mezičlánkem pak mohou být stavebnice určené zejména pro zábavu, kdy jsou obvykle dodány pouze brackety a servomotory a řídicí deska a software je nechán na volbě zákazníka. [1]

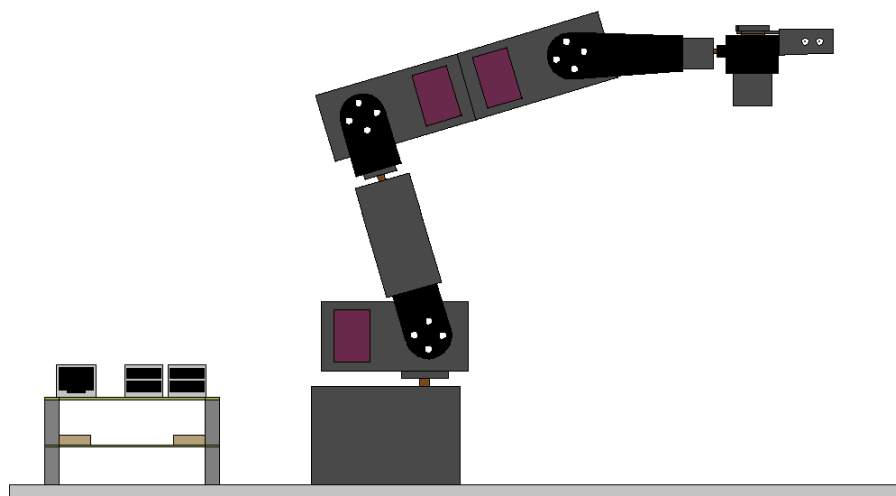
Jak je vidět z předchozí tabulky, všechny robotické ruce se pohybují vysoko nad hranicí čtyř tisíc korun, kdy tato cena obsahuje pouze samotnou ruku společně se softwarem. Je nutné dodat, že v tabulce obsahuje pouze kompletní stavebnice robotických rukou.

Touto prací bych tak chtěl dosáhnout toho, aby výsledný produkt byl konkurenceschopný cenově. Cílem je ovšem také dosáhnout toho, že robotická ruka společně s jejím softwarem bude jeden celek, tedy to, že nebude nutné používat další počítač pro její řízení. Tím je myšleno, že řídicí software bude mít současně GUI, které bude schopná řídicí deska

Stavebnice	Výrobce	Typ X	Řízení	Platforma	Kč
RA2-MINI	Arexx	Kompletní	Mikroprocesor	Windows	4000,-
RA1-PRO	Arexx	Kompletní	Mikroprocesor	C++	8000,-
C9895	Quasar	Kompletní	N/A	Windows	3500,-
AL5A	Lynxmotion	Kompletní	BotBoarduino	Windows	7500,-
Robotická ruka	Merkur	Kompletní	N/A	Windows	7000,-

Tabulka 5.1: Porovnání kompletních robotických rukou

zobrazovat na podporovaných typech monitorů. Cílem je tedy vytvořit robotickou ruku, která bude vypadat podobně jako na následujícím obrázku.



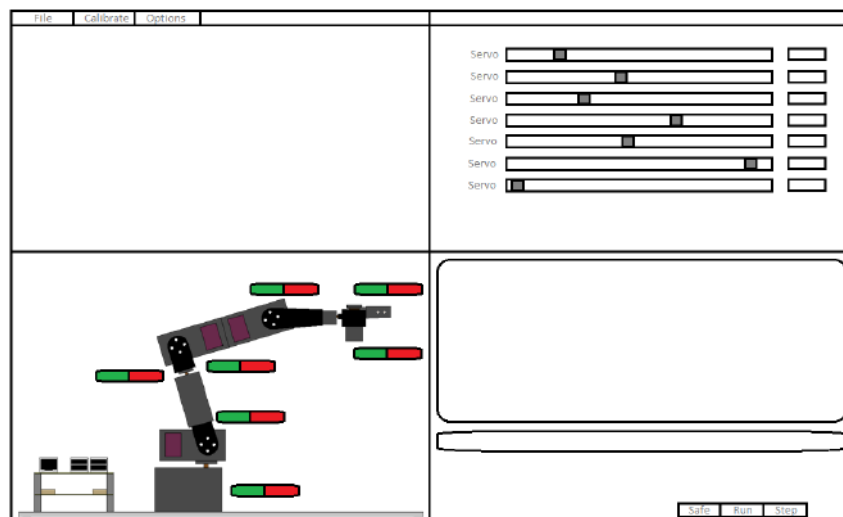
Obrázek 5.1: Návrh robotické ruky

Výsledkem práce by také měl být software, který bude robotickou ruku ovládat. Návrhem tohoto software je pak nutné adresovat tři nejčastěji užívané způsoby ovládání, které jsou v současné době dostupné. Těmi jsou ovládání pomocí posuvníků, funkce programování pomocí příkazů a zobrazení schématu robotické ruky společně s tlačítky pro ovládání. Posledním způsobem ovládání pak může být ovládání pomocí myši, kdy její poloha a kombinace stlačených tlačítek bude udávat polohu robotického ramene.

Výběr platformy a všech ostatních dílů samotné ruky probíhal v první řadě tak, aby byly jednotlivé díly dostupné ke koupi v České republice.

Pro řízení a jako samotný mozek celé ruky bude použito již dříve zmiňované a popisované Raspberry Pi ve své druhé generaci. Druhá generace byla vybrána zejména z hlediska vyššího výkonu vůči prvnímu Raspberry, které by se na daný účel nehodilo. Ze zpětného pohledu by se více hodilo Raspberry Pi 3, zejména z hlediska jeho vyššího výkonu. Problémem je však jeho energetická náročnost, kdy by nestačil standardně dostupný napáječ 5V a musel by být použit nějaký 12V napáječ a DC-DC konvertor.

Jako mozek celé ruky tedy posloužilo Raspberry Pi. Problémem však je, že řízení servomotorů z hlediska jejich signálů Raspberry zvládne zcela bez problémů, ovšem napájení samotných servomotorů je již na výstupní piny Raspberry Pi několikanásobně příliš. Obvyklý proudový výstup output pinu Raspberry Pi se pohybuje okolo hodnoty 25 mA. Je však možné z daného pinu dostat krátkodobě až 40 mA, to je ovšem stále málo na to, aby



Obrázek 5.2: Návrh GUI

bylo možno tyto piny použít na napájení servomotoru, který si při vytížení může říct až o 2A. [11] Je tedy nutné rozdělit napájecí a logické okruhy tak, aby bylo možno propojit signálové piny Raspberry Pi a zároveň napájecí piny servomotorů anižby bylo nutné složitě řezat a pájet dodávané kabely k servomotorům. K tomuto účelu byla navržena napájecí a signálová deska, obsahující všechny konektory pro připojení napájení k Raspberry Pi. Deska také obsahuje napájení jednotlivých servomotorů tak i propojení signálových (oranžových) konektorů a pinů na Raspberry Pi. Samotná deska bude popsána v sekci popisující jednotlivé části desky.

Jednotlivé servomotory jsou popsány dále v této kapitole. Pro samotné uchycení servomotorů do podoby jednotlivých kloubů ruky byly použity díly od společnosti Lynxmotion. Tato společnost se specializuje na výrobu a prodej robotických setů, kolových a pásových podvozků pro roboty, drony a jiné. Hlavní částí sortimentu, která bude využita, jsou však části pro vytváření kloubů. Popis jednotlivých částí je dostupný dále v této kapitole. Hlavní část celého ústrojí je základna, na kterou budou veškeré části celé ruky připevněny. Musí se jednat o dostatečně robustní a těžký materiál, který bude schopný vykompenzovat váhu ruky v jakékoliv poloze. Je možné použít ocelový plát tloušťky 2mm a rozměry 35x15cm.

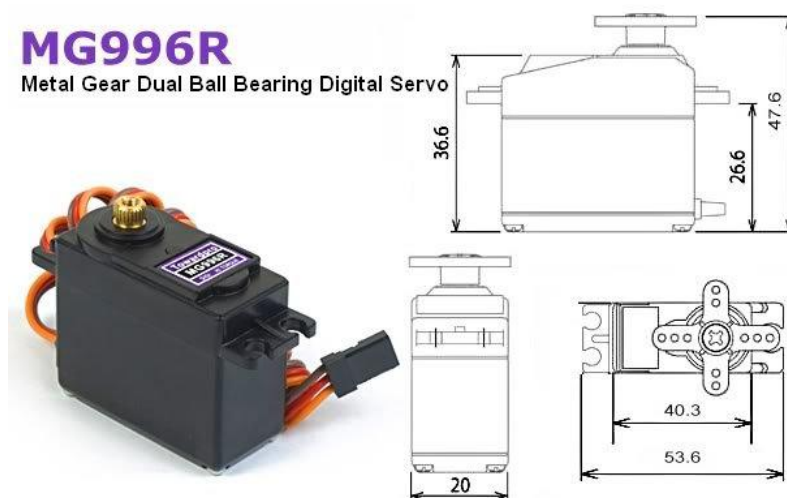
Poslední, ovšem neméně důležitá, část celé ruky je její napájení. Jak již bylo uvedeno o distribuci napětí se stará navrhnutá napájecí deska, ovšem tato deska je napájena standardním napáječem 5V/4A. V případě možnosti koupě výkonnějšího napáječe je toto lepší volba, jelikož 4A jsou zcela hraniční hodnotou pro spolehlivé fungování dané ruky.

### 5.1.1 Použité RC servomotory

Pro práci byly vybrány a použity tři typy RC servomotorů, zejména proto, že konečná serva u drapáku sloužící pro zavírání čelistí a otáčení čelistí nepotřebují disponovat takovou silou jako servomotory, které jsou použity pro vytvoření kloubů ruky.

Pro vytvoření kloubů ruky bylo potřeba vybrat, pokud možno, co nejsilnější standardní RC servomotory, ovšem s co nejlepším poměrem cena/výkon vzhledem k výsledné ceně celé ruky. Zároveň bylo dáno jasné kritérium, a to takové, že převody musejí být kovové, v opačném případě by ruka moc dlouho nevydržela. Volba tak padla na servomotory MG996R, které dle specifikace disponují kovovými převody a dvojími kuličkovými lo-

žisky. Jejich síla se pohybuje od 9,4 Kg/cm (4,8V) až po 11Kg/cm (6V), tedy při napětí 5V serva disponují silou kolem 10 Kg/cm. Zároveň se jedná o 120stupňové servo, které je schopné se pootočit o 60 stupňů za 0,16 s a tedy nebude omezovat pohyb ruky.



Obrázek 5.3: MG996R servomotor [<https://bit.ly/2rEPMJU>]

Pro použití jako otáčecí servomotor samotného drapáku se vzhledem k využití daného bracketu nehodil stejný servomotor jako na všechny ostatní klouby, tudíž bylo použito servo typu HS-81, které, ačkoli není přehnaně silné, pouze 2,6 kg/cm a disponuje pouze plastovými převody, tak vzhledem k jeho velikosti a rozšířenosti se perfektně hodí na danou pozici. V případě, že se plastové převody opotřebují, nebo se vzhledem ke špatné konfiguraci olámou, není problém koupit pouze samotné převody a poškozené převody vyměnit.



Obrázek 5.4: HS-81 servomotor [<https://bit.ly/2GeBlkk>]

Posledním použitým servomotorem je HS-322HD, který pro své použití v drapáku zcela dostačuje. Velikostně tento servomotor spadá do kategorie standardních servomotorů, s tahem 3 kg/cm. A ačkoli se jedná pouze o servomotor s karbonitovými převody, tak pro aplikaci zavírání a otevírání čelistí se velmi hodí, není možné, aby se s tímto výkonem čelisti zavřely příliš velkou silou a poškodily tak přenášený objekt.



Obrázek 5.5: HS-322HD servomotor [<https://bit.ly/2GeFpB8>]

Celkově tak pro tvorbu ruky bude potřeba 7 dříve popsanych servomotorů:

5x servomotor MG996R

1x servomotor HS-322HD

1x servomotor HS-81

Dále bude potřeba použít několik bracketů, které z jednotlivých servomotorů vytvoří funkční klouby. Seznam těchto dílů i s obrázky se nachází v příloze.

### 5.1.2 Návrh napájecí desky

Napájecí deska má kromě úlohy napájení Raspberry Pi a jednotlivých servomotorů také za úlohu propojit jednotlivé signálové kabely ze standardních servokonektorů na jednotlivé piny Raspberry Pi. Samotná napájecí část je řešena pouze pomocí 5V napáječe, takže není důvod přidávat jakékoliv měniče a usměrňovače. Jediná součástka této desky je 1000 uF elektrolytický kondenzátor, který má za úkol vyhladit krátkodobé špičky v odběru proudu servomotory tak, aby se zamezilo restartu Raspberry Pi z důvodu nedostatečného napájení.

Následně se na desce nacházejí už jen konektory propojující jednotlivé části celku tak, aby byla ruka pokud možno co nejvíce modulární. Z hlediska toho, že kabely od jednotlivých servomotorů v koncové části ruky (drapák apod.) jsou moc krátké na to, aby dosáhly až k napájecí desce, je nutné vložit mezikus, který tyto kabely prodlouží. V tomto případě byla využita jednoduchá redukce ze tří servomotorů na jeden signálový molex 3pin a napájecí 2pin.

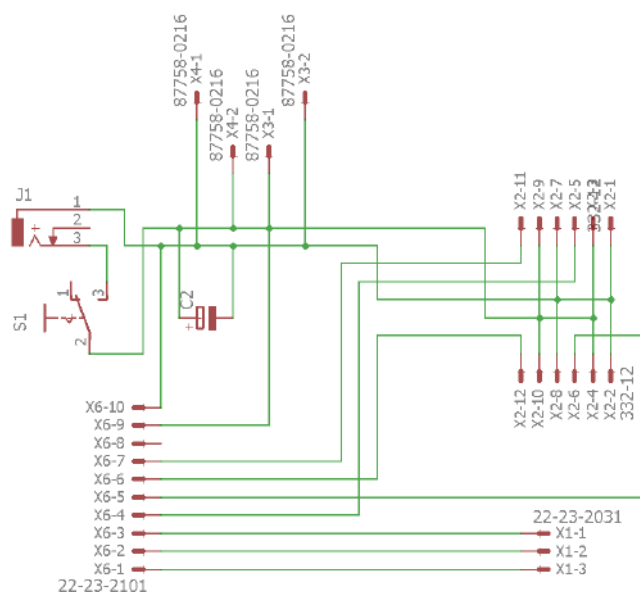
Dále je vyobrazeno samotné schéma této desky společně s návrhem PCB. V této verzi se zapnutí a vypnutí řeší pomocí jednoduchého vypínače, který spíná + větev napájení. Toto řešení je nejjednodušší možné, ovšem ve chvíli, kdy je spínač sepnut, sebou servomotory trhnou, toto trnutí je způsobeno chováním samotných servomotorů, které se takto projevují v případě, že servomotor nemá signál na svém signálovém konektoru. Tento problém by se dal vyřešit pomocí opto-článku, který by spínal napájecí + větev až ve chvíli kdy by bylo Raspberry Pi nabořováno. Toto řešení s sebou nese určitá úskalí, kdy by se musel přepracovat návrh napájení, jelikož by se musely rozlišit napájecí větve pro Raspberry a pro Servomotory. Ovšem nejedná se o zásadní problém, jeden pin na signálovém a napájecím konektoru k Raspberry Pi je stále volný (pin 8) a dal by se tak využít k tomuto účelu.



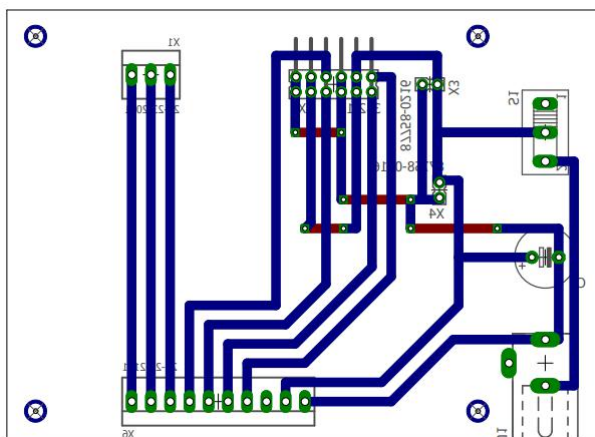
### 5.1.3 Návrh schématu a PCB

Při navrhování schématu napájecí desky bylo počítáno s jistou volností při propojování této desky a Raspberry Pi. Je to zejména proto, že ne všechny desky mají stejně rozmístěné piny a je tak nutné konzultovat zapojení signálových a napájecích konektorů přímo s datasheetem dané verze Raspberry.

V tomto případě jsou piny 9 a 10 použity jako napájecí piny pro Raspberry Pi a je tak nutné pin 9 připojit na pin 1 na Raspberry Pi (jedná se o + větev) a pin 10 připojit na pin 5 na Raspberry Pi (GND). Tímto je zajištěno napájení Raspberry Pi a jsou tedy propojeny kritické napájecí piny obou desek. Jelikož je napájení Raspberry Pi řešeno přímo přes GPIO piny, jsou obejity veškeré ochranné obvody Raspberry Pi, je tedy nutné dávat pozor na polaritu napájení. V opačném případě riskujeme poškození Raspberry.



Obrázek 5.6: Schéma napájecí desky



Obrázek 5.7: PCB napájecí desky

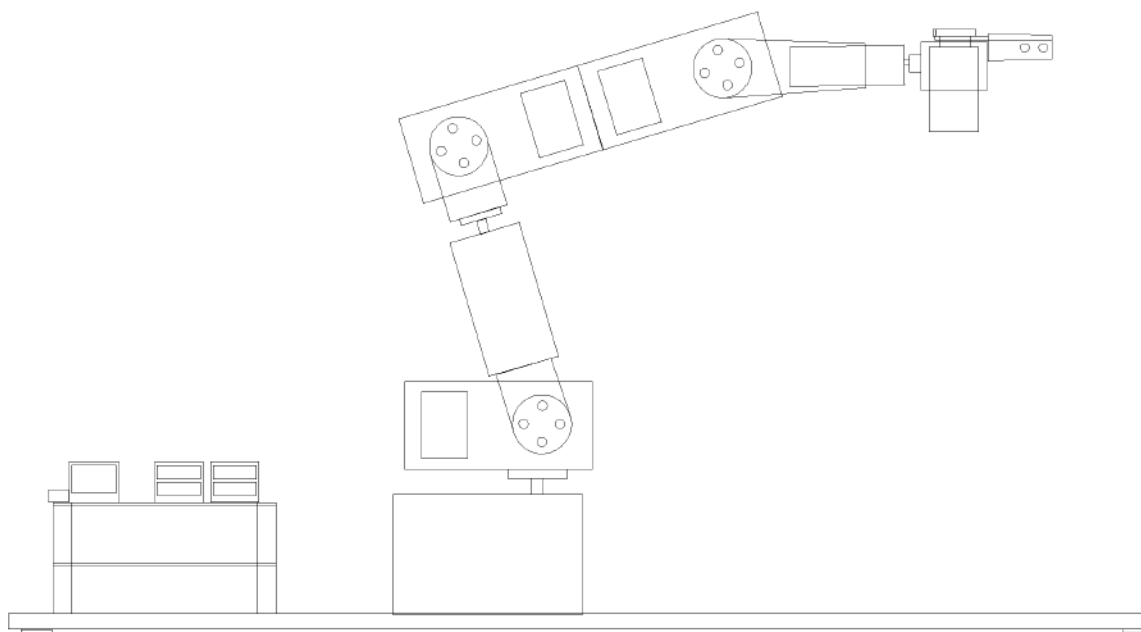


## 5.2 Sestavení robotické ruky

Sestavení robotické ruky bude probíhat zejména podle následujícího schématu toho, jak by ruka měla vypadat. Ruka sestává z celkem 6 kloubů plus jednoho drapáku. Prvním krokem při sestavování robotické ruky je pak vytvoření podložky, na které bude robotická ruka společně s řídicími obvody upevněna.

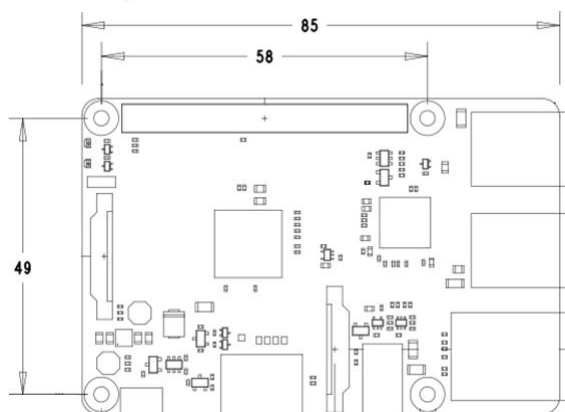
Tento popis se bude zabývat pouze základním popisem sestavení, kdy v případě, že by čtenář chtěl tuto robotickou ruku stavět, je podrobný návod přiložen v příloze. V tomto návodu jsou pak rozepsané veškeré klouby a jak je spojit do fungujícího celku.

Celý produkt je pak z hlediska sestavení možné rozdělit na dvě části. První částí je vytvoření napájecí desky a její spojení s Raspberry Pi a následně její uchycení k podložce. Druhou částí je pak sestavení a připevnění samotného ramene robotické ruky k podložce.



Obrázek 5.8: Schématický návrh robotické ruky

Při přípravě podložky je nutné počítat s nutností uchycení jednotlivých částí systému. Pro uchycení napájecí a řídicí desky byly zvoleny distanční podložky, které zabezpečuje dostatečný prostor mezi jednotlivými deskami. Pro uchycení těchto dvou komponent je do podložky nutné vyvrtat 4 otvory 3 mm vrtákem v rozměrech 58x49 mm znázorněných na dalším obrázku. V případě použití M3 šroubů je nutné zvětšit instalační otvory v Raspberry z rozměru M2.5 na rozměr M3. Tato změna není striktně vyžadována v případě, že použijeme šrouby M2.5, které jsou však mnohem hůře sehnatelné než zcela klasický rozměr M3, který je dostupný v každém železářství.



Obrázek 5.9: Raspberry Pi upevňovací otvory [<https://bit.ly/2liYAjm>] upraveno

Ve chvíli, kdy je napájecí i řídící deska připevněna k podložce, je možné připevnit hlavní bracket držící celé rameno robotické ruky připevněné k této podložce. K tomuto účelu slouží Multi Purpose servo Bracket, který je přichycen k podložce dvěma šrouby. Je nutné vybrat takové šrouby, které mají dostatečně plochou hlavu, tak, aby se do bracketu vešel i servomotor. Nabízí se tak použití zápusťných šroubů. Je nutné tento bracket umístit alespoň 3 centimetry od logické desky vzhledem k nutnosti zapojení konektorů jednotlivých servomotorů.

Při sestavování robotické ruky je nezbytné před samotným skládáním zajistit, že jsou všechny servomotory vycentrované. V případě, že je tento předpoklad zanedbán, se může stát, že bude servomotor upevněn v některé, z jeho maximálních pozic, kdy by tato chyba zamezila pohybu servomotoru a tím i celého kloubu v původně uvažovaném rozsahu daného kloubu. Pro toto centrování existuje několik možností. V případě již funkčního programu pro Raspberry Pi stačí pouze připojit servomotor do napájecí desky a mít propojené signálové piny. Pak jen stačí určit, který pin dané servo používá, a vycentrovat jej pomocí funkce „centre“.

V případě, že zatím není k dispozici funkční software pro řízení této robotické ruky, je nejjednodušší způsob centrování servomotoru pomocí servo testeru. Jedná se o malé zařízení, které je po připojení napájení schopné generovat ovládací signál pro servomotor. Tento servo tester obsahuje funkci vystředění, kdy stačí tuto funkci nastavit pomocí jediného „mode“ tlačítka.

Je také dobré vyzkoušet plný rozsah pohybu servomotoru, je tak zamezeno stavu, kdy je servomotor poruchový a tedy nemusí dosahovat obou svých hraničních hodnot nebo jeho pohyb nemusí být plynulý. Obvykle jsou tyto problémy způsobeny špatnou konektivitou mezi vnitřním potenciometrem a logikou servomotoru.



Obrázek 5.10: Servo tester [<https://bit.ly/2liO6vY>]

### 5.3 Prvky návrhu

Při sestavování robotické ruky je nutné dávat pozor u několika úkonů, kdy má jejich nesprávné provedení za následek buď nefunkčnost celého robotického ramena nebo jeho nepravdivý a trhaný pohyb.

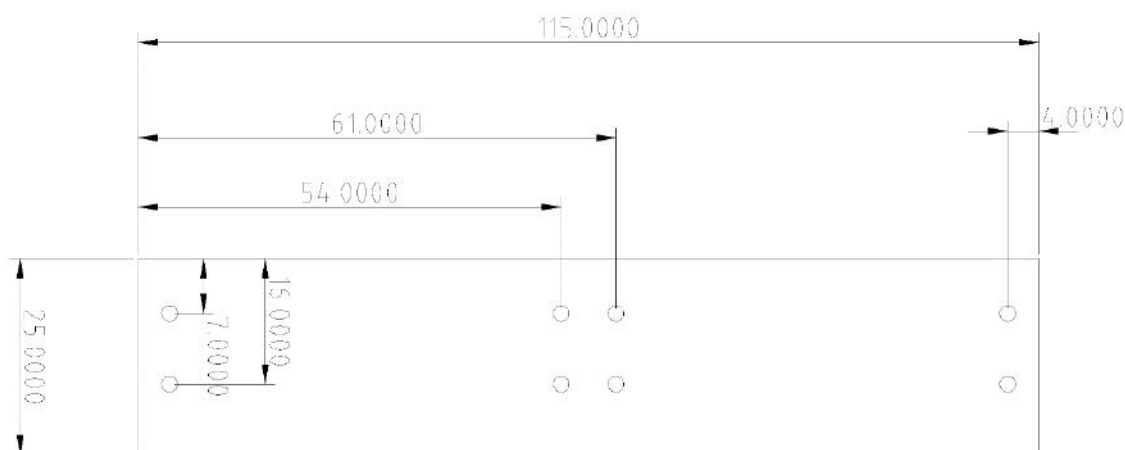
Prvním úskalím je správné dotažení servomotorové páky na jeho hřídeli. Toto je zejména nutné u kloubů jedna a tři, tedy hlavní otáčecí kloub a druhý otáčecí kloub. Tyto klouby jsou v ose, ve které se vyvíjí nejvyšší zatížení. Je tedy nutné, aby v těchto částech systému nebyla žádná vůle. V případě, že by byla vůle v hlavním otáčecím kloubu, se stane rameno nestabilní, což bude mít za následek nepravdivý a trhaný chod při zvedání a sklápění celé ruky.



Obrázek 5.11: Hlavní otáčecí kloub robotické ruky

Dalším úskalím pak může být koupě a vhodně zvolený bracket, tvořící poslední část ramene ruky. V tomto případě jsou servomotory připevněny tak, že jsou obě jejich osy vyvedeny směrem od sebe, je tedy nutné tyto servomotory pevně spojit. Za normálních okolností je možné využití Dual Inline Multi-purpose servo bracketu.

Je však možné, že tento bracket nebude možné koupit. V takovém případě je možné použití dvou multi purpose bracketů, které budou spojeny mezikusem, který je nutné vyrobit. Mezi servomotorem a bracketem je dostatečné místo pro vložení hliníkového profilu rozměrů 115x25 mm.



Obrázek 5.12: Návrh a rozměry hliníkového profilu

## 5.4 Zapojení servomotorů

Před zapojením servomotorů nejprve je nutno spojit napájecí desku s Raspberry Pi. Toho je dosaženo tím, že je propojen 10 pin na napájecí desce společně s 40pinovým konektorem na Raspberry Pi podle následujícího pin to pin schématu.

Pin napájecí desky (GND=PIN9)	Pin Raspberry Pi P1 header	Funkce
0	4	Servo 1
1	17	Servo 2
2	18	Servo 3
3	21	Servo 4
4	22	Servo 5
5	23	Servo 6
6	24	Servo 7
7	NC	NC
8	2	+5V
9	6	GND

Tabulka 5.2: Pin mapping v knihovně ServoBlaster

Rozložení pinů na Raspberry pi a napojení na odpovídající piny 10pinové patice na napájecí desce není kriticky důležité, jediné, co by se stalo při nesprávném napojení, je nutnost přepsání konfiguračního souboru pro ovládací software a namapování servomotoru na správný pin, který byl použit.

Jediné, na co je důležité dávat pozor, je pak použití GPIO pinů na Raspberry Pi. Použitá knihovna RPIO.PWM umí používat všechny GPIO porty. V případě použití knihovny Servo Blaster, která byla napsána pro první Raspberry Pi a je tak schopná používat pouze původní rozložení pinů na 26 pinové patici P1, je pro testovací účely důležité použít pouze piny s číslem menším než 25.

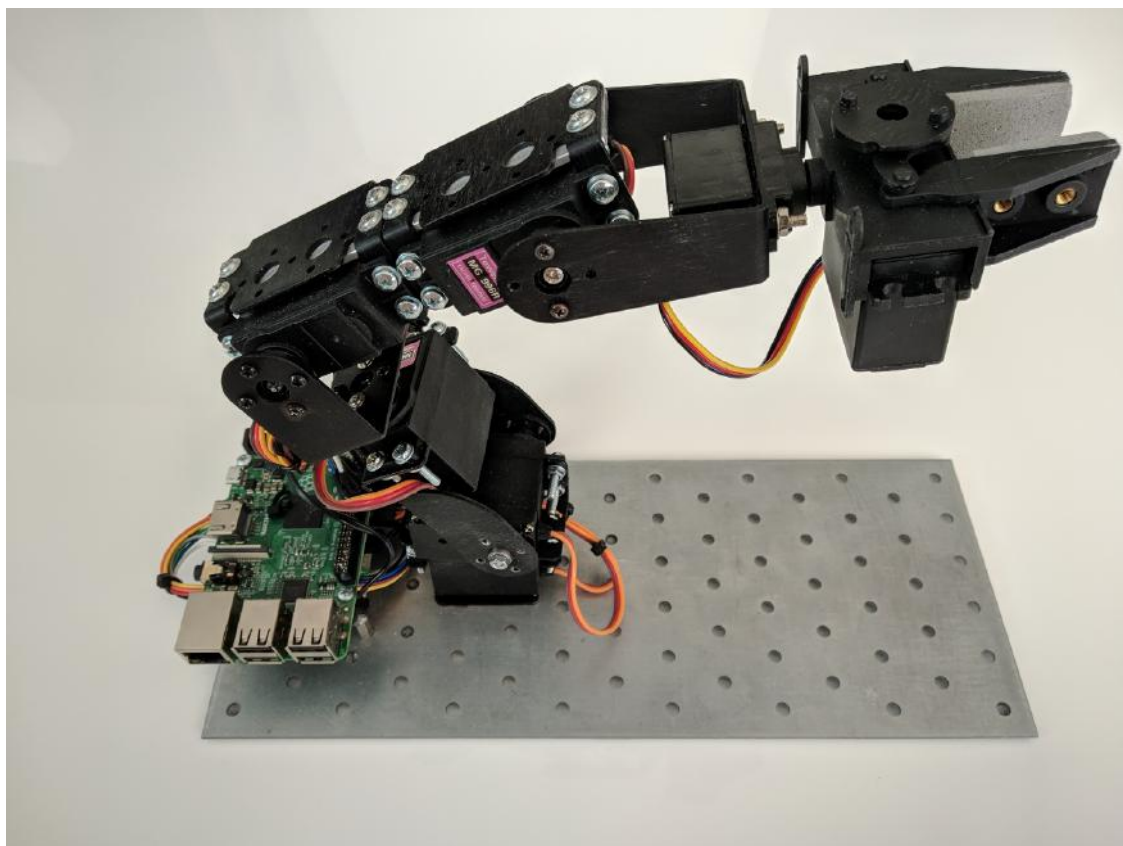
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I2C)		DC Power 5v	04
05	GPIO03 (SCL1 , I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Obrázek 5.13: Pinout Raspberry Pi [<https://bit.ly/2ImWQBe>]

## 5.5 Výsledný vzhled robotického ramene

Při skládání jednotlivých dílů k sobě je nutné dbát také na to, aby měly kabely dostatečnou vůli. V opačném případě pak existuje možnost, že se v lepším případě budou rozepínat konektory, v horším případě se pak mohou kabely vytrhnout z pájecích ploch v samotném servomotoru.

Ve chvíli, kdy však mají všechny kabely vůli a všechny klouby jsou spojené tak, jak mají, může z jednotlivých dílů vzniknout robotická ruka podobná této.



Obrázek 5.14: Sestavená robotická ruka

Jak je vidět, tak v levé části se nachází řídicí a napájecí deska, kdy přibližně tři centimetry od ní se nachází základna ramene robotické ruky. Je také vidět, že místo použití dual inline servo bracketu se použily dva klasické brackety spojené dříve popsaným hliníkovým profilem. Všechny kabely jsou pak vedeny spodní stranou, kdy je dbáno na co nejvolnější pohyb všech částí systému.

## Kapitola 6

# Software

Při tvorbě software k robotické ruce bylo cílem vytvořit software, který by mohl běžet přímo na ovladači této ruky, v tomto případě tedy Raspberry Pi. Nutností při návrhu bylo poskytnutí co nejvyšší volnosti uživateli.

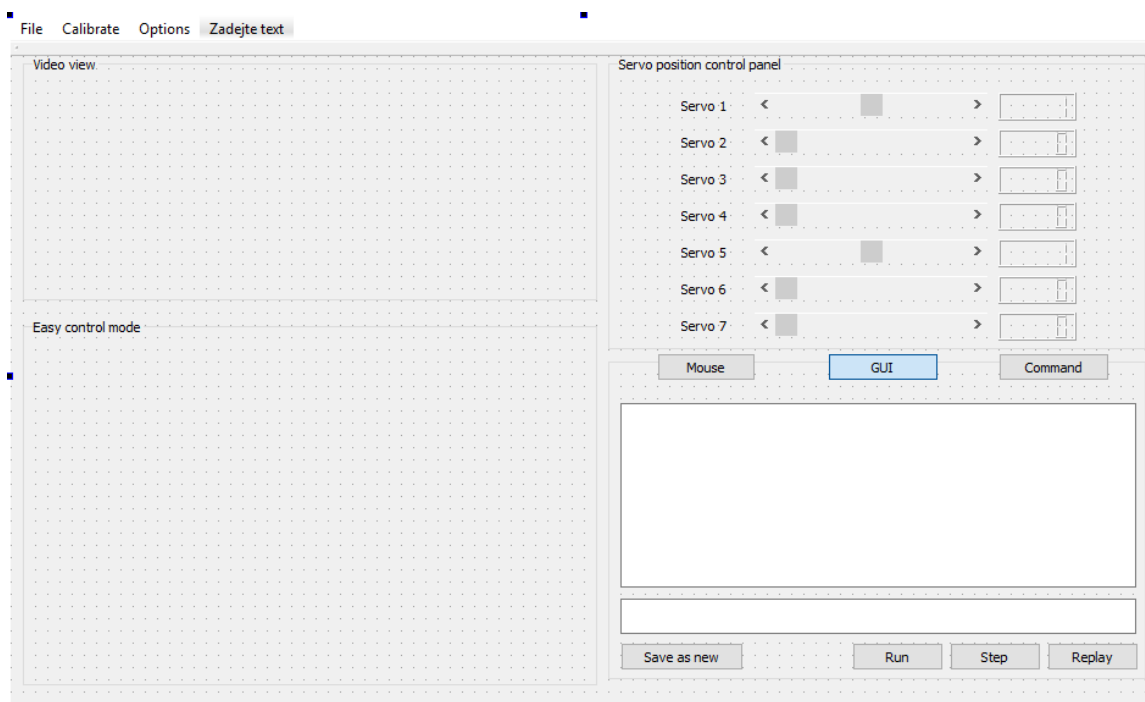
Tohoto cíle bylo dosaženo pomocí jednoduchého, avšak zcela zásadního nastavení, kterým je nastavení jednotlivých pinů pro servomotory a poskytnutí přístupu k nastavení jejich hraničních hodnot. Ačkoliv je počet servomotorů pevně omezen na 7, je možné tyto servomotory namapovat zcela podle preferencí uživatele. Je tedy možné daný software využít i pro jiné účely, než je pouze řízení robotické ruky, jako třeba jednoduchý ale účinný tester servomotorů.

### 6.1 Grafický návrh

Při návrhu grafického prostředí byl kladen důraz zejména na to, aby bylo jednoduché samotný software používat. Proto bylo grafické uživatelské prostředí rozděleno na čtyři bloky, kvadranty, které mají každý jiný způsob ovládání jednotlivých servomotorů.

V prvním kvadrantu bylo vyznačeno místo „Video view“, které by mělo sloužit pro zobrazování videa přenášeného z kamery na ruce. Tato vlastnost však nebyla do ruky v tuto chvíli implementována, a je myšlena spíše jako možné budoucí vylepšení současného stavu.

Druhý kvadrant „Easy control mode“ by měl obsahovat schématické zobrazení samotné robotické ruky společně s tlačítky, která dokáží ovládat specifický servomotor. Třetí kvadrant je vyčleněn ovládání servomotorů pomocí jednotlivých posuvníků a jedná se tak o přesné ovládání servomotorů. V posledním kvadrantu jsou zobrazena textová pole. Myšlenkou tohoto kvadrantu je určitý typ programování ruky.



Obrázek 6.1: Návrh GUI v QT Creatoru

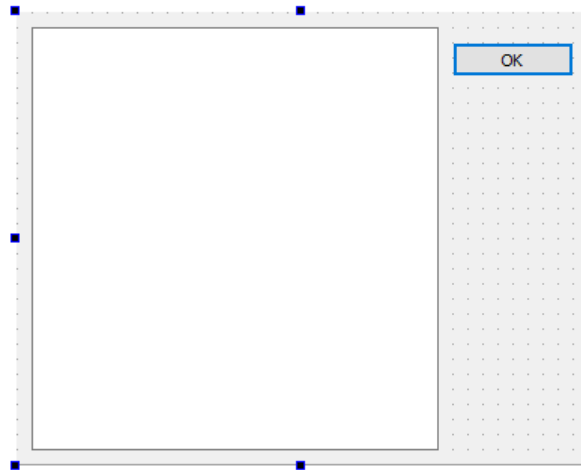
## 6.2 Grafické rozhraní

Jak již bylo v předchozí kapitole zmíněno, hlavní obrazovka je rozdělena do několika segmentů. Kromě těchto segmentů jsou tu i kontextové nabídky a tři hlavní tlačítka.

Kontextové nabídky File, Calibrate a Options mají za cíl shlukovat méně časté úkony pod jednu nabídku. Nabídka File tak obsahuje možnosti pro načítání a mazání příkazového souboru. Tento soubor se načítá ve čtvrtém kvadrantu hlavní obrazovky a obsahuje příkazy, které by ruka měla provádět.

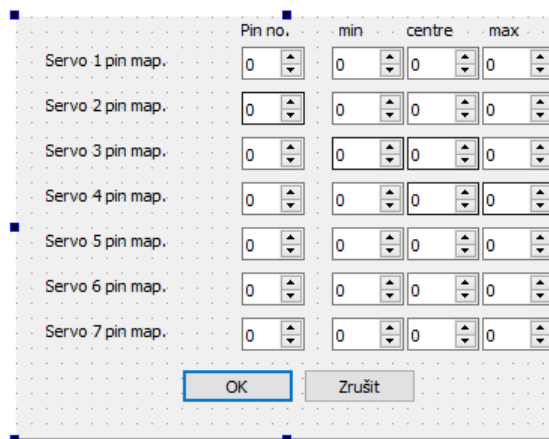
Nabídka Calibrate pak, jak již název napovídá, obsahuje tlačítka, která buď ruku natočí do základní dokovací pozice, a nebo naopak určí aktuální pozici jako novou dokovací pozici.

Poslední nabídka pak obsahuje tři podnabídky. První z nich je tlačítko Quit, které, jak již název napovídá, ukončuje celou aplikaci. Aplikaci je však možné ukončit i jiným způsobem, a to využitím tlačítka křížek na hlavním okně aplikace.



Obrázek 6.2: Návrh Help okna

Další dvě nabídky pak invokují nová okna, kdy první tlačítko Help vyvolá zobrazení okna s nápovědou a druhé tlačítko naopak vyvolá kontextovou nabídku pro změnu mapování jednotlivých tlačítek.



Obrázek 6.3: Návrh Pin mapping okna

Posledním prvkem, se kterým se může uživatel v tomto software setkat, je nabídka pro zadání souboru, ať už pro načítání souborů s příkazy, a nebo naopak při ukládání těchto souborů.



Obrázek 6.4: Návrh okna pro vložení pozice souboru



## 6.3 Struktura kódu

Grafické uživatelské rozhraní je samo o sobě psáno v Qt, přesněji v PyQt5, a o samotné ovládání jednotlivých servomotorů se stará knihovna RPIO.PWM. Obě tyto knihovny dokáží spolupracovat s programovacím jazykem Python, který tak tvoří samotný základ celého ovládacího software.

Celá struktura ovládacího software je rozdělena na několik částí. Hlavní částí je python script, který se stará o veškerou logiku celého software. Ovšem samotná logika nestačí a tak je nutné vytvořit grafické rozhraní. Výhodou Qt frameworku je jeho přenositelnost mezi platformami, nejedná se však o jeho hlavní výhodu. Výhodou je, že je schopný grafické rozložení celé obrazovky uložit do podoby .xml souborů popisujících vlastnosti jednotlivých prvků GUI. Následně je tak možné tyto soubory pomocí knihovny PyQt5 načítat přímo do python skriptu.

Tímto je zajištěno logické rozdělení samotného programu do částí podle jejich určení. Hlavní okno je tak načítáno ze souboru mainwindow.ui. Výhodou tohoto přístupu je možnost použití software, který se nazývá Qt Creator, díky kterému je možné rozložení daného okna vidět v reálném čase, aniž by bylo nutné cokoliv kompilovat.

## 6.4 Funkce software

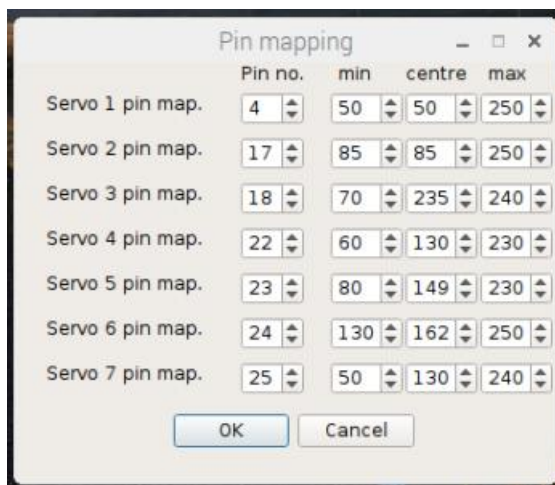
Jakmile je software spuštěn, jsou automaticky načteny defaultní hodnoty mapování pinů jednotlivých servomotorů. Tyto hodnoty jsou načítány ze souboru /Config/Pin.map obsahujícího jak mapování pinů, tak i jednotlivé maximální polohy servomotorů. Prostřední hodnoty udávají „centrální“ hodnoty servomotorů. Nejedná se ani tak o centrální hodnoty, jako spíš o hodnoty, které jsou použity pro základní dokování robotické ruky.

První řádek udává mapování jednotlivých pinů na servomotory, kdy jdou hodnoty vzestupně, tedy od servo 1 až po servo 7. Následují hraniční hodnoty jednotlivých servomotorů ve stejném pořadí.

```
1 4;17;18;22;23;24;25;  
2 50;50;250;  
3 85;85;250;  
4 70;235;240;  
5 60;130;230;  
6 80;149;230;  
7 130;162;250;  
8 50;130;240;
```

Obrázek 6.5: Mapping pinů v souboru

V případě, že není tento soubor přístupný, je použito defaultní mapování uvedené přímo v kódu a vypsán error při spouštění. Pokud je nutné změnit momentální mapování pinů nebo mapování hraničních hodnot přímo z GUI, je pro to přizpůsobena obrazovka dostupná v nabídce „Options“. Tato nabídka dovoluje změnit jakékoliv hodnoty po dobu běhu programu. Tím je docíleno toho, aby nedocházelo ke znefunkčnění celého software na základě špatně specifikovaného mapování. Je tak dobré před přepsáním konfiguračního souboru nejprve vyzkoušet požadované hodnoty přímo v GUI v této nabídce a až poté měnit konfigurační soubor.



Obrázek 6.6: Mapování pinů v GUI

Pro spuštění ovládacího software ruky jsou dvě možnosti. První z nich je použití .desktop souboru, který se na linuxových OS zobrazuje jako ikona. Tento soubor není závislý na svém umístění, je proto možné jej zkopírovat přímo na plochu. Při této volbě není invokován příkazový řádek, díky čemuž nejsou vypisovány pomocné výstupy.



Obrázek 6.7: Ikona softwaru

Druhou možností je spuštění přímo z příkazové řádky. Je tak nutné se přemístit do pracovní složky a použít příkaz „sudo python3.5 robotics.py“. Software je nutné spouštět se sudo právy, jelikož se pomocí něj přistupuje přímo do paměti. Je také nutné využít Python verze 3 a vyšší, jelikož nižší verze nemusejí stoprocentně fungovat.

Při prvotním spuštění se zobrazí nové okno software. Jsou aktivovány pouze dva módy, a to „Servo position control panel“ a „Easy control panel“. Druhý jmenovaný je aktivní za všech okolností, jelikož se jedná o nejjednodušší možné ovládání ruky.

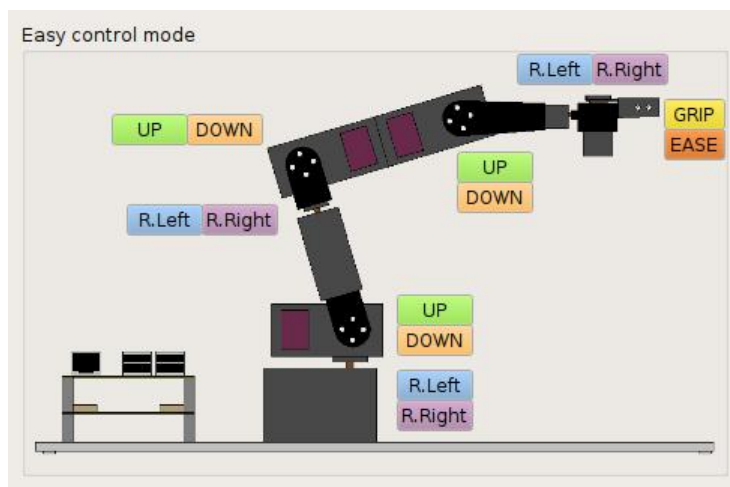
Jednotlivé módy se zapínají pomocí tlačítek uprostřed pravé strany obrazovky, kdy tlačítko „Mouse“ aktivuje pohybování ruky pomocí myši. Tlačítko „GUI“ pak aktivuje jednotlivé posuvníky. Tento mód je aktivní již při spuštění. Posledním módem je „Command“, který umožňuje spuštění příkazových souborů. Jednotlivé módy budou popsány níže.

#### 6.4.1 Easy Control mode

Prvním módem je Easy Control mode, který má za cíl intuitivně zobrazit a umožnit uživateli ovládání jednotlivých servomotorů, aniž by se musel zabýrat jejich pozicemi nebo si pamatovat, které servo je které.

Tento mód má pevně nastavenou rychlost pohybu, tou je pohyb o 1 kdy stlačení tlačítka zabezpečí tento pohyb (změna je možná v kódu). Je možné držet jednotlivá tlačítka stlačená

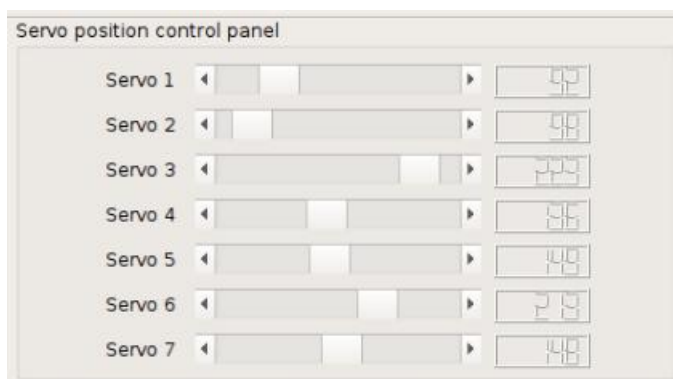
a tím dosáhnout plynulého pohybu. V případě dosažení maximální nebo minimální hodnoty, na kterou je servomotor schopný se přesunout, nebude již stlačení tlačítka mít další výsledek.



Obrázek 6.8: Easy Control Mode

Pro samotné fungování jak tohoto módu, tak i příkazového módu, je nutné udržovat poslední pozici servomotoru. Je to důležité zejména z důvodu toho, jak servomotor funguje. Servomotor, popřípadě knihovna, která ho ovládá, přijímá pouze velikost pulsu, který by měla vygenerovat. V případě, že bychom neznali původní pozici servomotoru, by bylo zcela nemožné posouvat servomotory o určitý počet stupňů.

#### 6.4.2 Ovládání pomocí posuvníků



Obrázek 6.9: Ovládání pomocí posuvníků

Na předchozím obrázku je pak vyobrazena další možnost ovládání servomotorů. V tomto módu můžeme servy pohybovat pomocí posuvníků. Samotné natočení jednotlivých servomotorů ve stupních je vidět na display vedle posuvníku. Pořadí servomotorů je možné určit pomocí mapování pinů.

Jednotlivé posuvníky je možné posouvat třemi způsoby. Prvním z nich jsou šipky na koncích posuvníků, druhým způsobem je chycení posuvníku pomocí myši a levého tlačítka a třetím je využití kolečka myši po najetí nad daný posuvník.

### 6.4.3 Command mode

V tomto módu se postupně procházejí jednotlivé příkazy zapsané v příkazovém souboru. Tento mód je aktivní pouze v případě, že je zmáčknuté tlačítko „Command“. Soubor se načítá z kontextové nabídky „File -> Command File“, která otevře dialogové okno pro zadání cesty k souboru.



Obrázek 6.10: Ovládání pomocí příkazů

Možnosti, jak tento soubor procházet, jsou dvě. První z nich dovoluje soubor přehrávat po jednotlivých příkazech. Toto má za cíl tlačítko „Step“. Naopak tlačítko „Run“ způsobí sekvenční procházení celého souboru. Posledním tlačítkem sloužícím pro práci se samotnými příkazy je pak tlačítko „Replay“, jehož stisknutí má za následek přesunutí čtecí hlavy na začátek souboru a tedy začne příkazy opět provádět od začátku. Jakmile čtecí hlava dojde na poslední příkaz, zastaví vykonávání dalších příkazů do doby, kdy není resetována pomocí tohoto tlačítka. Posledním tlačítkem je tlačítko „Save as new“, které vyvolá kontextovou nabídku pro zadání názvu nového souboru. Ukládání nemá příliš smysl v případě, že soubor pouze přehráváme, v případě však, že je soubor upravován, má již takto funkce své opodstatnění. Upravovat se dá přímo v textovém okně (velkém) malé okno slouží pouze pro zobrazení aktuálně prováděného příkazu.

V případě, že chceme příkazy upravovat, je jejich syntaxe následující: První pozice specifikuje, jaké servo se má hýbat, kdy číslo servomotoru odpovídá stejnému mapování jako u posuvníků, druhá pozice specifikuje cílový úhel natočení specifikovaného servomotoru. Třetí a poslední pozice specifikuje rychlost, s jakou se bude servomotor pohybovat. Jednotlivé pozice jsou odděleny středníky.

## Kapitola 7

# Testování

Testování obdobných systémů pro průmyslové použití probíhá zejména následovně. Při testování hardwarových součástí samotné robotické ruky se zkoumá zejména to, zda výsledný produkt odpovídá specifikaci, tedy zda vůle v jednotlivých součástech jsou v rámci specifikací, zda je naprogramovaný pohyb opakovatelný apod. Opakovatelnost je zásadním kritériem průmyslového robotického ramene. Zjišťuje se zejména, zda je naprogramovaný pohyb skutečně stále stejný při velkém počtu opakování, obvyklá tolerance v této kategorii se pohybuje okolo 0.2 mm. Testuje se také dosah ramene a hraniční hodnoty jednotlivých kloubů společně s plynulým chodem, důležitým kritériem je také maximální možná zátěž koncového dílu robotického ramene.

Software průmyslových robotických ramen je odlišný od softwaru používaného v domácích podmínkách. U průmyslového software je kladen nejvyšší důraz na spolehlivost. Jakákoliv chyba v softwaru totiž potenciálně může znamenat možnost poničení robotického ramene. Obvykle jsou tyto softwarové drivery vybaveny pouze programovací částí, kdy programování zůstává většinou totožné. Při průmyslovém použití je kladen důraz zejména na funkci.

Při testování podobných modelů těchto systémů se pak většinou testují jednotlivé maximální rozsahy jednotlivých kloubů, zda je chod dostatečně plynulý a zda odpovídá výsledný výrobek bezpečnostním předpisům pro možnost prodeje. V případě, že se jedná o stavebnici, je pak ještě obvykle otestováno, že je z dodaných dílů skutečně možné daný výrobek podle návodu sestavit.

### 7.1 Testování vytvořené robotické ruky

Testování výsledného produktu se dá rozdělit na dvě části. První je testování samotného hardware robotického ramene a druhou pak testování vytvořeného software.

Oproti klasickému testování však testování tohoto robotického ramene muselo probíhat opačně. To znamená, že nejprve proběhlo testování samotného grafického rozhraní, kdy byly zaznamenány požadavky a veškeré problémy daného návrhu a až pak se mohlo přistoupit k testování samotného hardware ruky. Toto pořadí je důležité zejména proto, že cílem bylo vytvořit obě části, tedy jak software, tak hardware. Problémem však je že bez funkčního software není možné správně otestovat funkčnost hardwaru ruky.

Cílem testování tak bylo zjistit, zda je vytvořený software dostatečně intuitivní, ale také dostatečně přizpůsobivý. Bylo nutné otestovat, zda všechny předem vytyčené cíle, které by měl software zvládnout, fungují, tedy zejména ovládání pomocí posuvníků, jednoduché

skriptování pohybu robotické ruky pomocí příkazů a ovládání pomocí tlačítek umístěných u jednotlivých kloubů robotické ruky. Posledním způsobem ovládání je pak ovládání pomocí myši, kdy bylo nutné otestovat, zda se jedná o dostatečně intuitivní možnost pro ovládání.

Hlavním cílem při testování hardwaru pak bylo zjistit, zda byly dodrženy počáteční požadavky na výsledný produkt. To znamená zejména nízký třes jednotlivých servomotorů, možnost zvedat břemena těžší než je 100 gramů ale také zjistit, zda je možné ruku replikovat a jak složité je její ovládání z hlediska jednotlivých způsobů ovládání.

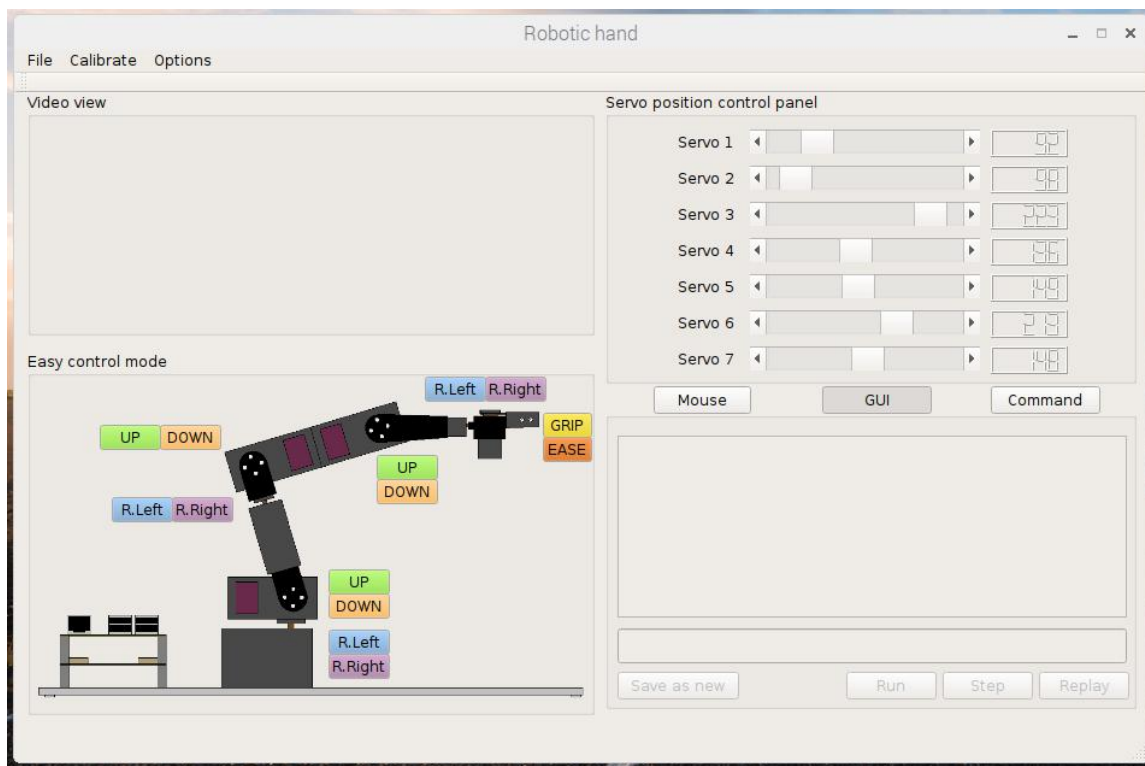
Testování obdobných systémů pro průmyslové použití probíhá ovšem mírně odlišně. Při testování hardwarových součástí samotné robotické ruky se zkoumá zejména to, zda výsledný produkt odpovídá specifikaci. Tedy zda jednotlivé vřely v jednotlivých součástech jsou v rámci specifikací, zda je naprogramovaný pohyb opakovatelný a podobně. Opakovatelnost je zásadním kritériem průmyslového robotického ramene. Zjišťuje se zejména, zda je naprogramovaný pohyb skutečně stále stejný při velkém počtu opakování, obvyklá tolerance v této kategorii se pohybuje okolo 0.2mm. Testuje se také dosah ramene a hraniční hodnoty jednotlivých kloubů společně s plynulým chodem, důležitým kritériem je také maximální možná zátěž koncového dílu robotického ramene.

Software průmyslových robotických ramen je odlišný od softwaru používaného v domácích podmínkách. U průmyslového software je kladen nejvyšší důraz na spolehlivost. Jakákoliv chyba v softwaru totiž potenciálně může znamenat možnost poničení robotického ramene. Obvykle jsou tyto softwarové drivery vybaveny pouze programovací částí, kdy programování zůstává většinou totožné. Při průmyslovém použití je kladen důraz zejména na funkci.

## 7.2 Testování GUI

Testování grafického rozhraní probíhalo pouze v rámci kamarádů a rodiny a zaměřovalo se zejména na rozložení a intuitivnost jednotlivých prvků ovládání. Na základě tohoto testování pak vzešel požadavek na barevná tlačítka pro Easy mode, díky čemuž je ovládání více intuitivní a člověk se spíše řídí pozicí tlačítka a jeho barvou než nápisem, který je zobrazen.

V původním návrhu také figuroval pouze schématický obrázek robotické ruky, tedy pouze tak, jak byl zobrazen na obrázku 5.8, což se však ukázalo jako krajně nevhodné řešení, jelikož na šedé podložce byl jen velmi těžko viditelný. Z tohoto důvodu se opět použily barvy pro kontrastní přechody.



Obrázek 7.1: Vytvořené kompletní GUI

Jak je vidět na předchozím obrázku, byla nakonec použita vybarvená verze schématického návrhu robotické ruky pro názornost umístění tlačítek pohybujiících jednotlivými servomotory. Na základě testování byla také přidána funkcionality odbarvení jednotlivých ploch, které nejsou používány. V případě tak, že není aktivní Command část jsou její tlačítka odbarvena. Je to zejména z důvodu, aby se předešlo zmatení uživatele při používání Easy Control mode a možnosti, že by považovali tlačítka Command módu za tlačítka náležící také k Easy Control módu.

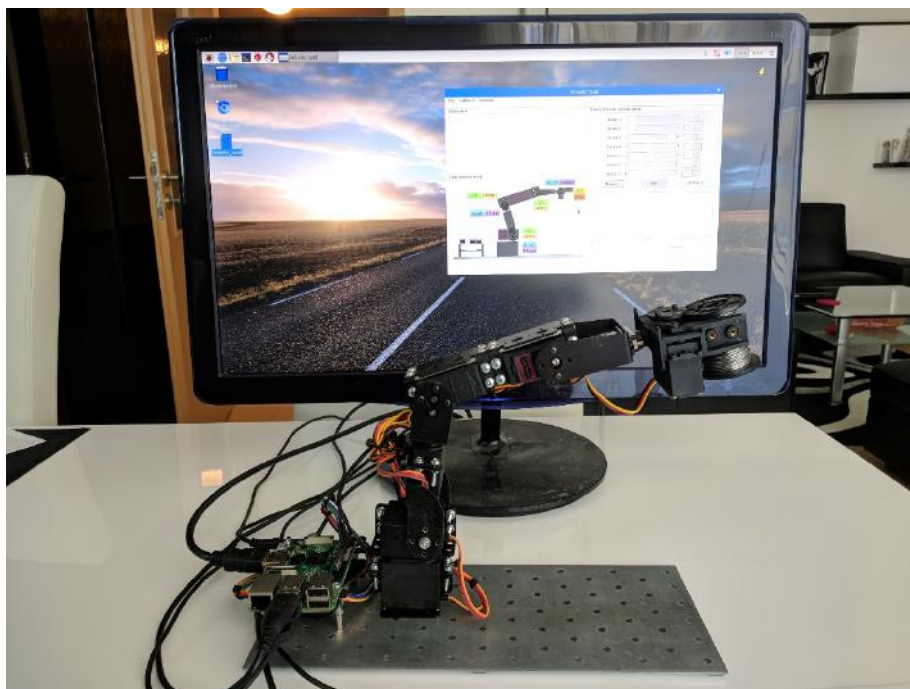
### 7.2.1 Testování Hardware

Testování hardwaru probíhalo stejně jako testování software, tedy výlučně mezi stejnou skupinou lidí. Za úkol bylo přemístit závaží, kterým pro účely testování byla špulka cínu, z bodu A do bodu B. Při tomto úkonu neexistovalo omezení na výběr metody, kterou bude robotická ruka ovládána, tester si tedy mohl zvolit jakýkoliv způsob. Ve druhé fázi však bylo nutné otestovat jednotlivé způsoby ovládání, a tak bylo nutné testery požádat o přesunutí závaží i pomocí ostatních typů ovládání.

Výsledkem tak bylo to, že posuvníky neposkytují dostatečné množství zpětné vazby, a tak bylo při používání nutné používat šipky na každé straně posuvníku pro jemný chod servomotorů. V tuto chvíli je však funkcionality těchto posuvníků na úrovni jednotlivých tlačítek v Easy mode. Hlavním přínosem posuvníků tak je možnost otestovat jednotlivé polohy pro skriptovací mód, kde jsou zadávány jednotlivé příkazy pro robotické rameno.

Jako dva nejlépe přínosné způsoby pohybu se pak ukázaly možnosti ovládání pomocí tlačítek a pomocí myši. Po vysvětlení kombinace jednotlivých tlačítek na myši a jaké důsledky tyto kombinace mají se jednalo o nejrychlejší způsob pro přemísťování závaží z místa

A do místa B. V případě použití tlačítek v GUI byla úspěšnost přesouvání stejná, ovšem značně pomalejší.



Obrázek 7.2: Robotická ruka společně s GUI

Při testování se však ukázaly dva zásadní problémy. Prvním z nich bylo, že uživatel potřebuje poměrně značný čas na to se s rukou seznámit natolik, aby omylem závaží spíše rukou neodhodil, než aby ho uchopil. Druhým, ale o to závažnějším problémem je, že ačkoliv bylo třepání servomotorů vyřešeno na softwarové úrovni tak hardwarová stránka je stále problematická. To je způsobeno použitím nepřesných servomotorů, díky čemuž se tyto servomotory vzhledem k páce celého ramene mohou pohnout, v tu chvíli však logika servomotoru pohyb vyrovná, jenže toto chování způsobuje třas. Tento problém se však vyskytuje pouze v některých polohách.



## Kapitola 8

# Závěr

Cílem práce bylo vytvořit robotickou ruku za použití standardních modelářských servomotorů, rozebrat způsob jejich ovládání a navrhnout software, který by byl schopný co nejsnazším způsobem tyto servomotory ovládat. Tento cíl byl splněn.

Prostudování způsobů řízení modelářských serv a jejich ovládání počítačem bylo vyřešeno v kapitole 2 Pohonné jednotky. Návrh jednoduché robotické ruky sestavené za pomoci modelářských serv a způsoby jejího ovládání byl proveden v kapitole 5 Cíle práce. Způsoby jejího ovládání pak byly rozebrány také v kapitole 2 Pohonné jednotky. Možnosti ovládání navrženého řešení a vyhodnocení vlastností v závislosti na použitých servech. Možnosti byly popsány v kapitole 3 Obvykle používané platformy a vyhodnocení vlastností pak proběhlo v kapitole 7 Testování. Navrženou ruku implementujte a demonstруйте její funkčnost. Této části bylo také dosaženo v kapitole 5 Cíle práce, potažmo pomocí návodu v příloze. Diskutování dosažených výsledků a možnosti pokračování práce jsou pak rozebrány dále v tomto textu.

V práci bylo zpracováno generování přesně stanovené šířky PWM signálu pro řízení servomotorů. K tomuto bylo nutné využít DMA přenosy, jelikož klasické způsoby generování PWM signálů pomocí procesoru nedostačovaly pro úplné zabránění samovolného třepání se servomotorů. Proto je lepší použití DMA knihovny, kdy nedochází k nekonzistenci velikosti výsledného pulsu vzhledem k jeho generaci pomocí procesoru. Tato metoda je použita na připraveném software, který je tak použitelný pro téměř jakýkoliv typ servomotoru. Je to díky tomu, že je možné nastavit každému servomotoru jeho maximální a minimální hodnoty (jedná se o šířku pulsu). Jakékoliv třepání servomotorů způsobené špatnou velikostí signálu je tak vyřešeno. V některých polohách hlavního zvedacího servomotoru se můžeme s tímto problémem stále setkat. Příčinou toho však není signál, ale nekvalitní elektronika samotného servomotoru, kdy vnitřní potenciometr není dostatečně citlivý. Tento problém se dá vyřešit použitím kvalitnějších servomotorů. Zásadní problémy při sestavování a návrhu této ruky se nevyskytly. Hlavním problémem tak bylo především pevné spojení jednotlivých kloubů ruky tak, aby nedocházelo k jejich viklání.

V budoucnosti bych chtěl robotickou ruku přestavět za použití kvalitnějších servomotorů, které by netrpěly kvalitativními nedostatky, které následně způsobují nedostatečně plynulý chod celého ramene. Ve chvíli, kdy budou použity silnější a zároveň kvalitnější servomotory, je pak robotické rameno schopné plynulého pohybu, jak je ostatně vidět u posledních dvou HITEC servomotorů. V případě budoucí práce na této robotické ruce se jeví jako možné vylepšení přidáním kamery, která by byla schopná snímat podložku přímo pod čelistmi robotické ruky. Toto vylepšení pak přináší možnosti snímání této podložky a detekci předmětů, jako i řízení této robotické ruky na základě těchto dat.



# Literatura

- [1] *6 DOF Robotic Arm Kit - With Servo Motors - Unassembled*. [Online; navštíveno 10.3.2018].  
URL <https://www.thingbits.net/products/6-dof-robotic-arm-kit-with-servo-motors-unassembled>
- [2] *ArmUno 2.0 Robotic Arm*. [Online; navštíveno 10.3.2018].  
URL <http://microbotlabs.com/robot-kits.html>
- [3] *Atmel / SMART SAMA5D3 Series*. [Online; navštíveno 10.12.2017].  
URL [http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11269-32-bit-Cortex-A5-Microcontroller-SAMA5D3-Xplained\\_User-Guide.pdf](http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11269-32-bit-Cortex-A5-Microcontroller-SAMA5D3-Xplained_User-Guide.pdf)
- [4] *Jak funguje servo*. [Online; navštíveno 9.11.2017].  
URL <https://www.rc-zoom.cz/jak-funguje-servo/>
- [5] *RobotGeek Snapper Arm V3 Getting Started Guide*. [Online; navštíveno 10.3.2018].  
URL <http://learn.robotgeek.com/getting-started/33-robotgeek-snapper-robot-arm/63-robotgeek-snapper-arm-getting-started-guide.html>
- [6] *The Robot arm C-9895 CEBEKIT*. [Online; navštíveno 10.3.2018].  
URL <http://www.farnell.com/datasheets/815416.pdf>
- [7] *ROBOT ARM - Documentation SOFTWARE WEBSITE*. 2010, [Online; navštíveno 10.3.2018].  
URL [http://www.arexx.com/robot\\_arm/html/en/documentation.htm#manual](http://www.arexx.com/robot_arm/html/en/documentation.htm#manual)
- [8] *Servo Motor – Types and Working Principle*. 2015, [Online; navštíveno 9.11.2017].  
URL <https://www.electronicshub.org/servo-motors/>
- [9] *RaspberryPI models comparison*. updated 2018, [Online; navštíveno 13.12.2017].  
URL <http://socialcompare.com/en/comparison/raspberrypi-models-comparison>
- [10] *What is a Pulse Width Modulation (PWM) Signal and What is it Used For*. updated 2018, [Online; navštíveno 14.11.2017].  
URL <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA00Z00000190kFSAU>
- [11] Ada, L.: *Introducing the Raspberry Pi 2 - Model B*. 2015, [Online; navštíveno 1.5.2018].  
URL <https://cdn-learn.adafruit.com/downloads/pdf/introducing-the-raspberry-pi-2-model-b.pdf>

- [12] Boloor, S. M., Adith Jagdish: *Arduino*. Birmingham, United Kingdom: Birmingham : Packt Publishing, 2016, ISBN 9781787124530.
- [13] Bolton, D.: *7 Top Python GUI Frameworks for 2017*. 2017, [Online; navštíveno 10.1.2018].  
URL <https://insights.dice.com/2017/08/07/7-top-python-gui-frameworks-for-2017-2/>
- [14] Chroboczek, M.: *Grafická uživatelská rozhraní v Qt a C++*. Brno : Computer Press, 2013, ISBN 978-80-251-4124-3.
- [15] Crowder, R. M.: *Electric Drives and Electromechanical Systems: Applications and Control*. New York, USA: Elsevier Science, 2006, ISBN 0750667400.
- [16] Douglas, V. H.: *Microprocessors and interfacing programming and hardware*. Boston, USA: Glencoe McGraw-Hill, 1991, ISBN 978-0070257429.
- [17] Ellis, G. H.: *Basics of the Electric Servomotor and Drive*. New York, USA: Elsevier Inc., 2012, ISBN 978-0-12-385920-4.
- [18] Ghirst, R.: *ServoBlaster*. 2015, [Online; navštíveno 17.11.2017].  
URL <https://github.com/richardghirst/PiBits/tree/master/ServoBlaster>
- [19] Grusin, M.; Byron, J.: *Hobby servo tutorial*. [Online; navštíveno 9.11.2017].  
URL <https://learn.sparkfun.com/tutorials/hobby-servo-tutorial>
- [20] Hager, C.: *RPIO*. 2013, [Online; navštíveno 15.12.2017].  
URL <https://github.com/metachris/RPIO>
- [21] Hager, C.: *RPIO.PWM, PWM via DMA for the Raspberry Pi*. 2013, [Online; navštíveno 10.11.2017].  
URL [https://pythonhosted.org/RPIO/pwm\\_py.html](https://pythonhosted.org/RPIO/pwm_py.html)
- [22] Horan, B.: *Practical Raspberry Pi*. New York : Apress, 2013, ISBN 978-1-4302-4971-9.
- [23] Kingsley-Hughes, A.: *10 alternatives to the Raspberry Pi*. 2016, [Online; navštíveno 10.12.2017].  
URL <https://www.zdnet.com/pictures/10-alternatives-to-the-raspberry-pi/>
- [24] Laird, C.: *GUI Programming in Python*. 2018, [Online; navštíveno 8.1.2018].  
URL <https://wiki.python.org/moin/GuiProgramming>
- [25] Patidar, V.: *Survey of Robotic Arm and Parameters*. International Conference on Computer Communication and Informatics, 2016, ISBN 978-1-4673-6680-9.
- [26] Salt, J.: *Understanding RC Servos*. [Online; navštíveno 12.11.2017].  
URL <https://www.rchelicopterfun.com/rc-servos.html>
- [27] Univerzita, M.: *Knihovny pro tvorbu grafického obsahu*. [Online; navštíveno 6.1.2018].  
URL [https://is.mendelu.cz/eknihovna/opory/zobraz\\_cast.pl?cast=29058](https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=29058)
- [28] Williams, A.: *Understanding DMA*. 2017, [Online; navštíveno 10.11.2017].  
URL <https://hackaday.com/2017/02/28/understanding-dma/>

- [29] Zahra, S.: *RC Servo Control*. [Online; navštíveno 7.11.2017].  
URL  
<https://sites.google.com/site/controlandelectronics/rc-servo-control>

## Příloha A

# Návod na sestavení

### A.0.1 Použité brackety

1x Lynxmotion Little Grip



Obrázek A.1: Drapák [<https://bit.ly/2rI0f6C>]

1x Lynxmotion Micro servo rotate bracket



Obrázek A.2: Micro servo rotate bracket [<https://bit.ly/2rI0f6C>]

2x Lynxmotion Multi-purpose Servo Bracket



Obrázek A.3: Multi-purpose Servo Bracket [<https://bit.ly/2rI0f6C>]

2x Lynxmotion „C“ Servo Bracket



Obrázek A.4: "C" Servo Bracket [<https://bit.ly/2rI0f6C>]

1x Lynxmotion Dual Inline Multi-purpose Servo Bracket



Obrázek A.5: Dual Inline Multi-purpose Servo Bracket [<https://bit.ly/2rI0f6C>]

1x Lynxmotion Servo Offset Bracket



Obrázek A.6: Servo Offset Bracket [<https://bit.ly/2rI0f6C>]

Různý instalační materiál v podobě M3 šroubů, podložek, matek, stahovacích pásků a podobných není nutné uvádět z důvodu možnosti použití různých typů.

### A.0.2 Použitý napáječ

1x Napáječ 5V/4A



Obrázek A.7: 5V/4A napáječ [<https://bit.ly/2L1jQrs>]

### A.0.3 Použitý mozek robotické ruky

1x Raspberry Pi 2 model B 1GB ram

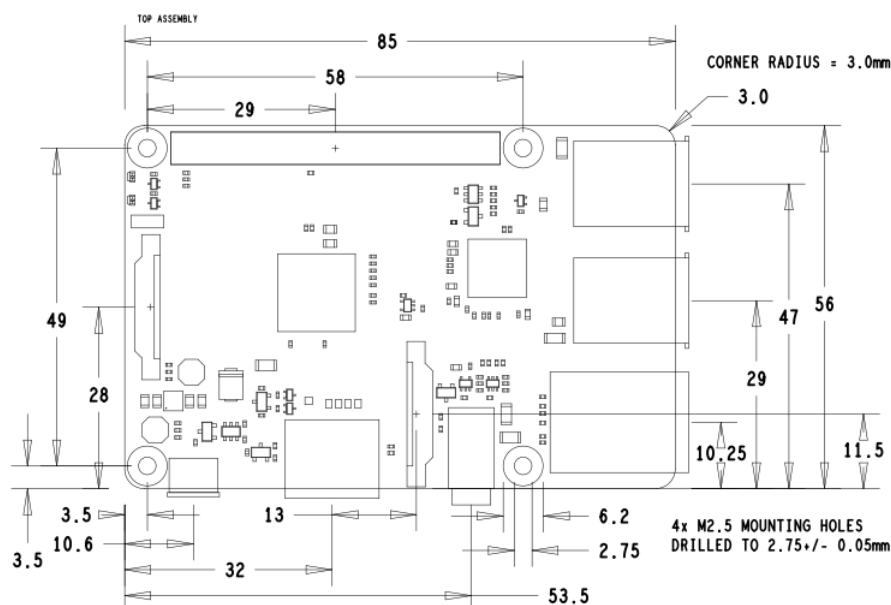


Obrázek A.8: Raspberry Pi 2 model B 1GB ram [<https://bit.ly/1m1YfSJ>]

### A.0.4 Postup sestavení

V rámci uchycení obou desek společně je nutné mít k dispozici dostatečný počet stand-off podložek v rozměrech 4x 10 mm a 4x 20 mm. Menší použijeme pro uchycení napájecí desky do základny a delší následně pro přichycení a zároveň uchycení Raspberry Pi. Delší stand-offy používáme zejména z důvodu přítomnosti elektrolytického kondenzátoru na napájecí desce.





Obrázek A.9: Raspberry Pi instalační otvory [<https://bit.ly/2IiYAjm>]

Jakmile je přichycena napájecí deska společně s Raspberry Pi je možno připevnit základní Lynxmotion Multi-purpose Servo Bracket, který bude držet hlavní otáčecí kloub ruky připevněný k podložce. Pro uchycení tohoto bracketu je opět nutné vyvrtat 2 M3 díry do základny. Pro přichycení tohoto bracketu je však nutné vybrat šrouby s co nejnižší hlavou. Doporučoval bych použití zápusťných šroubů společně s použitím zápusťného vrtáku tak, aby hlava šroubu příliš nevystupovala z bracketu. V případě, že tohoto nebude dosaženo bude obtížné do bracketu uchytit servomotor. Výběr pozice následně určí pozici hlavního kloubu ruky. Je dobré umístit tento bracket kolem 3 cm od napájecí desky vzhledem k nutnosti manipulace s kabely od jednotlivých servomotorů.

### A.0.5 Jednotlivé klouby

Cílem je seznámit se se způsobem a sestavením jednotlivých kloubů ruky tak, aby jednotlivé klouby fungovaly co možná nejlépe.

### A.0.6 Hlavní otáčecí kloub

Jak již bylo zmíněno, tento kloub je napevno připevněn k podložce. K tomuto účelu se využívají dva multi purpose brackety, které mezi sebou mají jeden servomotor. Jakmile je připevněný první bracket k podložce je možné do něj vložit a připevnit první servomotor. Tento servomotor se uchytí pomocí 4 M3 šroubů, dále je potřebovat 8x podložku a 4x matku. Při sestavování všech kloubů je použito pořadí hlava-podložka-servomotor-bracket-podložka-matka tak, aby bylo dosaženo pevného uchycení servomotoru a zároveň se vyvarovali problémům s měkkostí plastového těla servomotoru. V případě, že je v bracketu málo místa je možné toto servo vypodložit dalšími čtyřmi podložkami, kdy každou podložkou získáme asi 0,2 mm místa.



Obrázek A.10: Raspberry Pi instalační otvory

Následně je na hřídel servomotoru nutné připevnit páku. V tomto případě budeme používat kruhovou hřídel s tisícíhranem, která byla dodána společně se servomotorem. V případě, že existuje možnost získání kovové části místo plastové, je to lepší volba.



Obrázek A.11: Raspberry Pi instalační otvory

Jakmile je nasazena páka na hřídeli, je nutné ji utáhnout dodávaným středovým šroubem. Tato část je kriticky důležitá, jelikož v případě, že nebude daný aretační šroub dostatečně utažen, tak nebude tento kloub pevný a bude se viklat. V takto kritickém bodě ovšem vyklání znamená, že celá ruka bude nestabilní. Šroub je velikosti M3 a v případě, že by i přes veškerou snahu nebylo spojení zcela pevné, je možné použití delšího šroubu společně s podložkou. Je ale nutné si dát pozor, zda již šroub není moc dlouhý a neutahujeme závit až do plastového těla servomotoru. V takovém případě je nutné použití kratšího šroubu. Obvyklá délka je 10 mm. Jakmile je tisícíhran přichycen, je možné k němu připevnit čtyřmi samořeznými šrouby další bracket, který bude držet první servomotor, starající se o naklápění celé ruky.



Obrázek A.12: Raspberry Pi instalační otvory

### A.0.7 Druhý kloub

Jakmile je připevněný druhý bracket na prvním servomotoru (předchozí krok), můžeme připevnit další servomotor. Tento servomotor má funkci naklápění celé ruky, je to tedy nejnamáhanější kloub celé ruky. Samotný servomotor je schopný unést celou váhu robotické ruky plus náklad, který je přenášen.



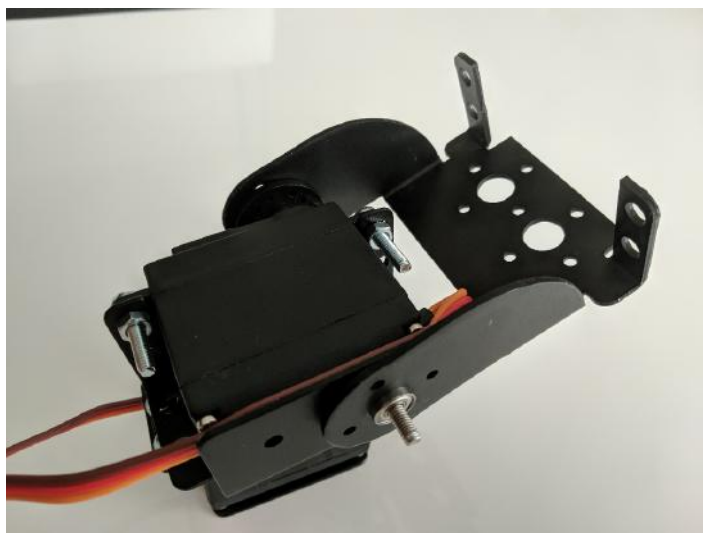
Obrázek A.13: Raspberry Pi instalační otvory

Při vytváření tohoto kloubu bude potřeba Servo Offset Bracket, který je důležitý proto, jelikož má delší rameno, a je tak schopný vytvořit dostatečné místo pro pohyb servomotoru. V krajní poloze, kdy je ruka nejvíce zvednutá, bude další servomotor spočívat na tomto servomotoru.

Je to z důvodu toho, že v této poloze bude ruka nejčastěji, a je tak možné „odložit“ další servomotor na tomto servomotoru. Zajistíme tak to, že ve chvíli, kdy bude ruka v dokovací poloze, tak nebude servomotor starající se o naklápění ruky pod žádnou pákou. Je tak zajištěn klidný chod tohoto servomotoru. V případě, že má servomotor signál, se bude vždy snažit udržet požadovaný úhel natočení. Čím větší páka na tento servomotor

působí, tím problematictější bude tento úhel udržet, a je možné, že servomotor začne mírně oscilovat. Toto chování je způsobeno nepřesností potenciometru v samotném servomotoru, podle kterého vnitřní logika určuje úhel natočení. Problémem tedy je, že jakmile si vnitřní logika uvědomí, že se servomotor pootočil více, než by podle signálu měl, bude se snažit vykompenzovat tento pohyb protiběžným pohybem a dostat se tak na požadovanou hodnotu úhlu natočení. V tu chvíli se toto chování může jevit jako oscilace mezi dvěma body.

Přichycení Servo Offset Bracketu probíhá stejně jako upevnění jakéhokoliv jiného bracketu. Jediným rozdílem je, že tento bracket je uzpůsoben obousměrnému uchycení. Je tedy nutné využít dodávaného ložiska. Je kritické neopomenout na to, že je nutné před upevněním servomotoru do bracketu naproti hřídeli servomotoru vložit šroub, který se bude chovat jako hřídel otáčení. Tento šroub je vidět na následujícím obrázku. Všimněme si, že ložisko je uchyceno drážkou uvnitř, tím je docíleno toho, že ložisko nevyklouzne. Takto připravený šroub s ložiskem je nutné zajistit maticí nebo dodávanou plastovou maticí. Je nutné mezi maticí a ložisko vložit podložku, která zajistí plynulý chod a správnou funkci ložiska.



Obrázek A.14: Raspberry Pi instalační otvory

### A.0.8 Třetí kloub

Třetím kloubem je kloub otáčející rukou v její ose doprava a doleva. Toto chování má za následek větší volnost pohybu a také větší a širší dosah samotné ruky. Pro tento kloub platí stejné požadavky na uchycení tisícíhranu jako pro první otáčecí servomotor. U zbylých kloubů je tento požadavek zanedbán, jelikož pohyb probíhá v ose servomotoru a je tak jedno, jak moc utažený tisícíhran bude. V tomto případě je však osa servomotoru v ose s pákou celé ruky. Osa tohoto servomotoru je tak více namáhána a je důležité, aby bylo toto spojení co možná nejpevnější. Jak je vidět na obrázku na tento tisícíhran pak následně bude připevněn „C“ type Bracket, který bude sloužit jako spojení hlavní a také nejdelší části ruky (paže).

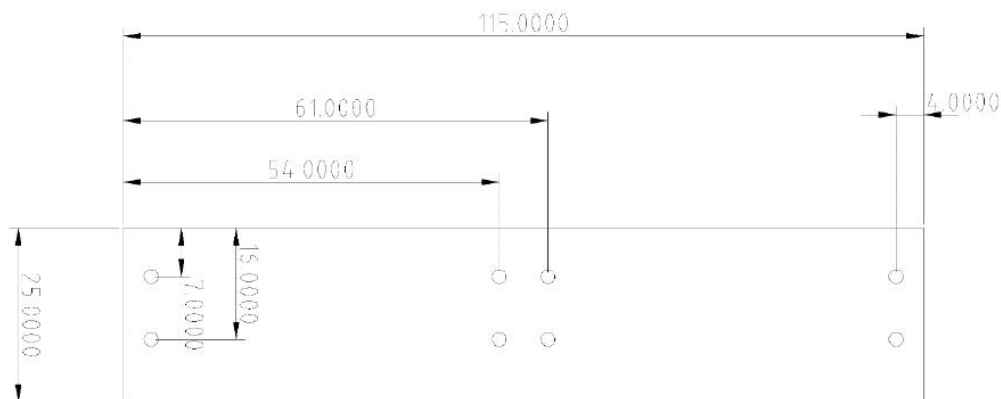


Obrázek A.15: Raspberry Pi instalační otvory

#### A.0.9 Čtvrtý a pátý kloub ruky

V tuto chvíli je seskládán spodní hardware ruky. V tuto chvíli je sestavována hlavní paže ruky. Tato paže sestává z dvou servomotorů MG996R a Dual Inline Multi-purpose Servo Bracket tak, aby oba servomotory měly svoji osu na každé straně tohoto bracketu.

V případě, že není možné sehnat tento prodloužený bracket, je možné použít dva standardní brackety, které ovšem uvnitř budou spojeny pomocí pomocného hliníkového profilu. Hliník je použit z důvodu jeho váhy a pevnosti. Do samotného bracketu se mezi bracket a servomotor vejde profil tloušťky 2 mm, který je zcela dostačující pro dané použití. Okótované rozměry a vzhled tohoto profilu je vidět na následujícím obrázku.



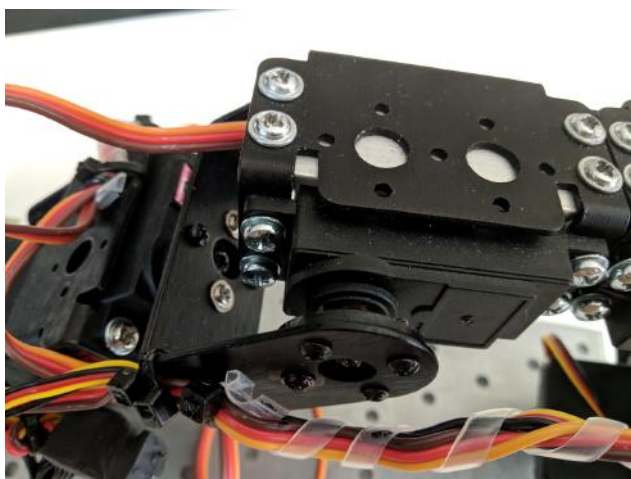
Obrázek A.16: Raspberry Pi instalační otvory

Další postup je již pro obě varianty stejný, je nutné uchytit oba servomotory do bracketu tak, aby jejich osy směřovaly směrem od sebe. Díky tomu získáme 115 mm paži, díky které se zvýší dosah ruky a zároveň se díky tomu, že jsou pro ohyb ruky použity prakticky 3 servomotory, zvedne ohebnost celé ruky.



Obrázek A.17: Raspberry Pi instalační otvory

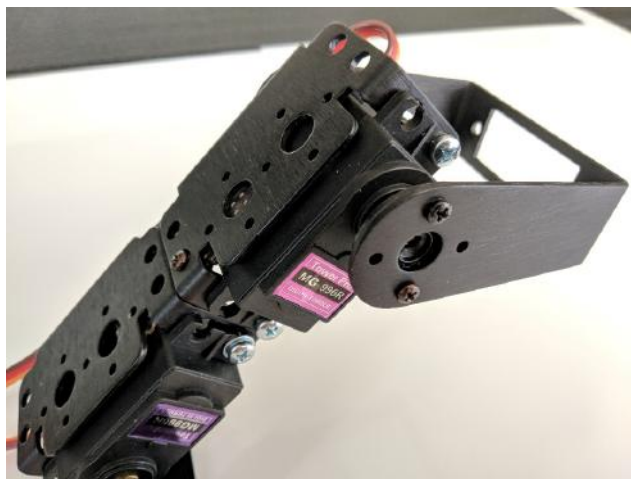
Na dané osy je opět nutné připevnit tisícíhrany a následně připevnit tuto paži k „C“ bracketu vedoucímu ze servomotoru číslo 3.



Obrázek A.18: Raspberry Pi instalační otvory

#### A.0.10 Čtvrtý kloub – Kloub drapáku

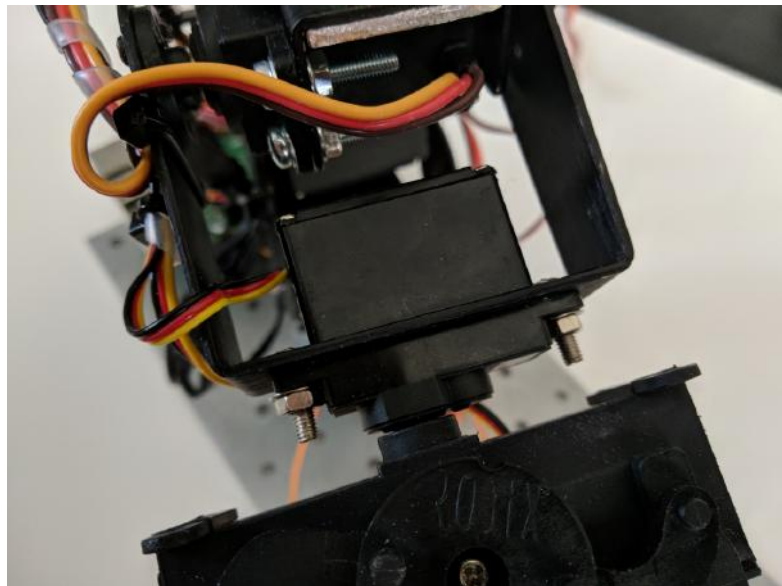
Na druhý konec dříve vytvořeného ramena přijde připevnit Micro servo rotate bracket, který bude sloužit jako přechodka a redukce pro další servomotor. Před upevněním bracketu je nutné zjistit, zda se HS-81 vejde do před-vyřezané díry. Je možné, že díra bude nedokonale vytvořená a bude nutné pomocí pilníku vypilovat rohy tak, aby se servomotor vešel.



Obrázek A.19: Raspberry Pi instalační otvory

Jakmile je známo, že se servomotor do daného otvoru vejde, je možné připevnit bracket k servomotoru. Je důležité použít servomotor HS-81, popřípadě obdobný servomotor stejné velikosti. Ovšem je nutné zachovat výrobce Hitec, jelikož každý výrobce využívá jiný rozměr tisícíhranu. Obvykle je rozdíl v počtu zubů. V případě servomotorů, které jsou uchyceny pomocí tisícíhranu, který k nim byl dodán, není tento problém podstatný, jelikož se použije dodaný tisícíhran.

V tomto případě však bude servo přímo namontováno na drapák, který je již vyliisovaný s určitým tisícíhranem a není možné jej vyměnit.



Obrázek A.20: Raspberry Pi instalační otvory



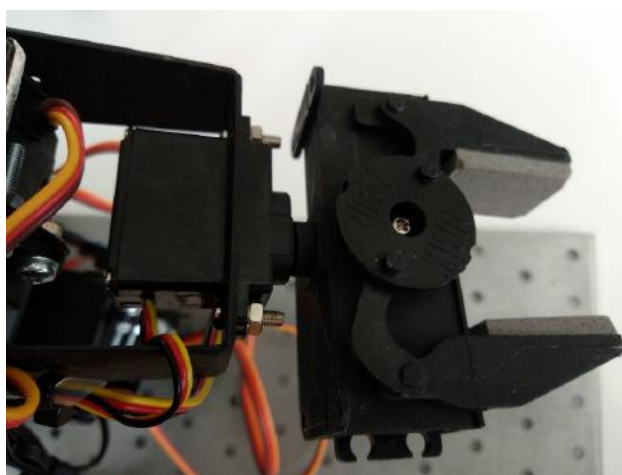
### A.0.11 Drapák

Posledním dílem celé ruky, ale také nejdůležitějším, jelikož by bez něj nebylo možné nic uchopit, je drapák. Ten je připevněný k servomotoru HS-81 v ose s pákou ruky, je tedy opět nutné dotáhnout dostatečně aretační šroub na ose servomotoru.



Obrázek A.21: Raspberry Pi instalační otvory

Sestavování probíhá v pořadí, kdy se nejprve drapák nasadí na servomotor a následně se pomocným otvorem mezi čelistmi prostrčí šroubovák tak, aby bylo možné dotáhnout aretační šroub. V případě, že bychom nejprve vložili servomotor svírající čelisti, pak by nebylo možné tento šroub dotáhnout.

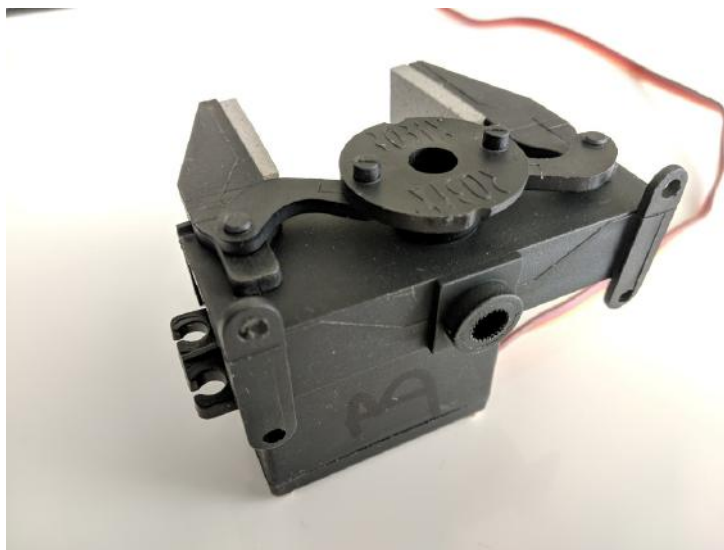


Obrázek A.22: Raspberry Pi instalační otvory

Samotný servomotor pohybuující čelistmi není nijak uchycen k tělu drapáku. Jediné, co tento servomotor drží na místě, je tření mezi stěnou drapáku a stěnou servomotoru plus



aretační šroub. Opět platí, že použitý servomotor musí být Hitec nebo taková kopie, která využívá stejný počet zubů v tisícihranu.



Obrázek A.23: Raspberry Pi instalační otvory

#### A.0.12 Instalace software

Pro fungování tohoto software je nutné doinstalovat dva moduly pro jednotlivé knihovny. Samotný Python3 není potřeba instalovat, jelikož je součástí image systému, stejně jako většina standardních knihoven pro python.

##### RPIO.PWM

První knihovnu, kterou je nutné nainstalovat je RPIO.PWM. Tato knihovna slouží pro řízení servomotorů a je tak kriticky důležitá. Standardní způsob instalace je pomocí pip nebo easyinstall, problémem je, že tyto způsoby obsahují pouze originální verzi této knihovny ve svém repozitáři, díky čemuž daná knihovna funguje pouze na Raspberry Pi první generace.

Proto abychom obešli tento problém, je nutné nainstalovat tuto knihovnu přímo ze zdrojů, tedy buď naklonovat repozitář z Githubu pomocí gitu, a nebo si celý repozitář (<https://github.com/metachris/RPIO>) stáhnout jako zip a rozbalit. Jakmile je repozitář rozbalený, je nutné spustit instalační skript, toho je dosaženo pomocí příkazu „sudo python3 setup.py install“. V případě, že by bylo nutné danou knihovnu provozovat i pomocí předchozí verze pythonu, stačí při instalaci použít pouze python místo python3.

##### PyQt5

Instalace PyQt5 je o poznání jednodušší, jelikož se poslední verze této knihovny nachází v apt-get repozitářích. V tuto chvíli tak stačí nainstalovat tuto knihovnu jako jakýkoliv jiný software na linuxový stroj. Je důležité nejprve provést „apt-get update“ a následně „apt-get install qt5-default pyqt5-dev pyqt5-dev-tools“, kdy je každý balíček důležitý pro správnou funkci celé knihovny. Tyto příkazy nainstalují základní moduly této knihovny, jako je QtCore, QtGui a QtWidgets. V případě, že by bylo potřeba více modulů pro další dodatečné úpravy software, pak je nutné tyto zbývající moduly doinstalovat.

## Ovládací software

Posledním prvkem při instalaci je pak zprovoznění samotného ovládacího programu, tedy software, který byl popsán v minulé kapitole. Jeho instalace je jednoduchá, za předpokladu, že byly obě potřebné knihovny nainstalovány tak, jak bylo popsáno, neměl by být problém se spuštěním programu, ať už pomocí .desktop souboru, nebo přímo pomocí příkazu „sudo python3 robotics.py“ v kořeni adresáře.